

ENHANCED MULTI-RESOLUTION ANALYSIS FOR MULTI-DIMENSIONAL DATA UTILIZING LINE FILTERING TECHNIQUES

MATTHEW J. PICKLO*^{†‡} AND JENNIFER K. RYAN*^{†§}

Abstract. In this article we introduce Line Smoothness-Increasing Accuracy-Conserving Multi-Resolution Analysis (LSIAC-MRA). This is a procedure for exploiting convolution kernel post-processors for obtaining more accurate multi-dimensional multi-resolution analysis (MRA) in terms of error reduction. This filtering-projection tool allows for the transition of data between different resolutions while simultaneously decreasing errors in the fine grid approximation. It specifically allows for defining detail multi-wavelet coefficients when translating coarse data onto finer meshes. These coefficients are usually not defined in such cases. We show how to analytically evaluate the resulting convolutions and express the filtered approximation in a new basis. This is done by combining the filtering procedure with projection operators that allow for computational implementation of this scale transition procedure. Further, this procedure can be applied to piecewise constant approximations to functions, as it provides error reduction. We demonstrate the effectiveness of this technique in two and three dimensions.

Key words. Discontinuous Galerkin, post-processing, SIAC filtering, Line filtering, accuracy enhancement, error reduction, Multi-Resolution Analysis

AMS subject classifications. 65M60

1. Introduction. Approximating multi-scale phenomena accurately and efficiently has its significance illustrated in such areas as turbulence modeling and kinetics among many other applications. Of considerable importance is the ability to match data on a coarse grid with that on a fine grid. Multi-resolution analysis (MRA) is one technique that is useful in accomplishing such a task [2, 11, 1, 22]. MRA expresses the approximation in terms of averages and differences, with the differences being the details between a fine grid approximation and coarse grid approximation. The MRA approximation usually starts at the finest grid and moves to a coarser grid. The reverse is a challenging procedure as the difference coefficients that allow for higher levels of resolution do not exist. In this paper, we introduce a procedure which allows for approximating the difference coefficients as well as error reduction when moving to a finer grid. To accomplish this, we introduce Line Smoothness-Increasing Accuracy-Conserving multi-resolution analysis (LSIAC-MRA). As the name suggests, it relies on the Line SIAC filter found in [10] that is an improvement over the tensor-product SIAC filters [6, 9, 20]. Previously, SIAC filters were used for one-dimensional MRA in [17]. We present the necessary operators and algorithms to accomplish this task as well as demonstrating its effectiveness on two- and three-dimensional functions. Aside from demonstrating that LSIAC-MRA can reduce the errors with mesh refinement, we show that it is also effective on piecewise constant approximations.

Multi-resolution Analysis (MRA) is a useful technique for moving data between successive mesh refinements. Initially introduced in the 1980s by Meyer and Mallat [21, 25], the MRA framework was extended to discrete data representations by Harten [1] and expanded with the addition of multi-wavelets by Alpert [2]. MRA-based wavelet techniques have found application in a variety of numerical methods for partial differential equations [3, 7, 11, 22, 24]. A survey of wavelet-based methods in fluid mechanics can be found in [14]. MRA is based on the idea that an approximation can be represented as a linear combination of scaling functions, which are typically the approximation basis, and wavelets. The scaling functions give the averages of the approximation over a coarse grid and the wavelet information contains the information necessary to move to a finer grid. Mathematically, the expression is the following:

$$(1.1) \quad \underbrace{u_h^f(x, t)}_{\text{fine grid approximation}} = \underbrace{u_h^c(x, t)}_{\text{coarse grid information}} + \underbrace{\sum_{j=1}^N \sum_{k=0}^p d_{k,j}^N(t) \psi_{k,j}^c(x)}_{\text{details necessary to move to a finer grid}} .$$

In this article, we use the basis functions for the multi-wavelet space given by Alpert, $\psi_{k,j}^c(x)$. These multi-wavelets are associated with the piecewise orthonormal Legendre approximation given in [2]. In this

*Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO 80401, USA.

[†]Research supported by Air Force Office of Scientific Research (AFOSR), Computational Mathematics Program (Program Manager: Dr. Fariba Fahroo), under grant number FA9550-20-1-0166.

[‡]Email: mpicklo@mines.edu

[§]Corresponding Author. Email: jkryan@mines.edu

expression, both the scaling functions and wavelets are piecewise polynomials of degree p . We expect that this procedure can be extended to generalized wavelets.

In creating this variant of Nyström reconstruction [16], we utilize the LSIAC filter [10] for multi-dimensional data. Nyström reconstruction also uses convolution for accurate reconstructions. For LSIAC-MRA, it allows for reducing the computational cost of the multi-dimensional filtering operation by introducing a one-dimensional support for the convolution kernel.

Pairing Smoothness-Increasing Accuracy-Conserving (SIAC) post-processing with the multi-resolution analysis presented in [2, 11] is natural. Although SIAC has broad applicability, it has mostly been developed for discontinuous Galerkin (DG) methods, which rely on a piecewise polynomial approximation basis. An introduction to this pairing in one-dimension can be found in [17]. However, the multi-dimensional equivalent is challenging as discussed in Section 4. Therefore in order to get to this stage we discuss the necessary background in multi-resolution analysis (Section 2.2), followed by Line SIAC filters (Section 2.3) before introducing LSIAC-MRA as a means to approximate difference coefficients. Demonstration of the effectiveness of this technique can be found in Section 5.

2. Background. In the following sections we present the ideas necessary for understanding enhanced multi-resolution analysis for multi-dimensions. To better illustrate the ideas, we begin with a discussion of the one-dimensional ideas covered in [17].

2.1. Input Data Format. For ease of discussion, we will assume our data is given in a modal format. This generalizes to the nodal format by simply assuming that the approximation basis is the Lagrange polynomial basis. Therefore, we describe the procedure for constructing projected approximations for higher dimensional quadrilateral and hexahedral meshes under consideration. In the following, we discuss the definitions necessary to implement LSIAC-MRA.

For a given function $u(\mathbf{x})$, we wish to construct a piecewise-polynomial approximation $u_h(\mathbf{x})$. To do so, we utilize the following notation: A multi-index $\alpha = (\alpha_1, \dots, \alpha_n)$ is an n -tuple of non-negative integers. Define $|\alpha| = \sum_{i=1}^n \alpha_i$, and $\partial^\alpha = \partial^{\alpha_1} \dots \partial^{\alpha_n}$, where $\partial^{\alpha_i} = \frac{\partial^{\alpha_i}}{\partial x_i^{\alpha_i}}$. Let $\mathbf{x} = (x_1, \dots, x_d)$. Consider a d -dimensional rectangular domain and its partition into N^d d -dimensional rectangular elements: $\Omega = \cup_{\beta \in \mathbb{B}} \mathcal{T}_\beta$, where the indexing set is given by

$$\mathbb{B} = \{\beta \in \mathbb{N}^d : \|\beta\|_{\ell^\infty} \leq N\}.$$

Utilizing the form of a discontinuous Galerkin type approximation, we define an approximation space over our partition that will contain piecewise polynomials in d variables up to a given degree p :

$$V_h^p = \{v \in L^2 : v \in \mathbb{P}^p(\mathcal{T}_\beta), \beta \in \mathbb{B}\},$$

where h is typically associated with the mesh size. We approximate $u(\mathbf{x})$ via L^2 -projection, that is on a given element \mathcal{T}_β , we write

$$u_h(\mathbf{x}) \Big|_{\mathcal{T}_\beta} = \sum_{\|\alpha\|_{\ell^\infty} \leq p} u_\alpha^\beta \phi_\alpha^\beta(\mathbf{x}), \quad \langle u, \phi_\alpha^\beta \rangle_{\mathcal{T}_\beta} = \langle u_h, \phi_\alpha^\beta \rangle_{\mathcal{T}_\beta},$$

where $\{\phi_\alpha^\beta\}_{\|\alpha\|_{\ell^\infty} \leq p}$ is a basis for our approximation space on \mathcal{T}_β . Note the standard approximation space notation above conflicts with the Multi-resolution analysis notation to be used below. For our purposes we equate V_h^p and V_n^p when $N = 2^n$.

2.2. Multi-Resolution Analysis. Multi-resolution analysis (MRA) introduced in [21, 25] provides a framework in which to analyze modal approximations under scale transition. Specifically, MRA views approximations as belonging to a hierarchy of nested approximation spaces, where transition between scales is simply the addition or removal of finer-detail basis functions called multi-wavelets. To better express the utility of this setting as it relates to our modal projections, we provide a brief summary of MRA as detailed in [17]. More in depth introductions to multi-resolution analyses generated by scaling functions can be found in [2, 3, 4, 7]. For ease of presentation, the one-dimensional case is given.

Consider a nested sequence of approximation spaces

$$V_0^p \subset V_1^p \subset \dots \subset V_n^p \subset \dots,$$

where each approximation space is given by

$$V_n^p = \{v : v \in \mathbb{P}^p(I_j^n), j = 0, \dots, 2^n - 1\},$$

with

$$I_j^n = (-1 + 2^{-n+1}j, -1 + 2^{-n+1}(j + 1)), \quad j = 0, \dots, 2^n - 1.$$

Denote by $\{\phi_k(x)\}_{k=0}^p$ the orthonormal basis on the coarsest mesh consisting of one element, $n = 0$. These basis functions are called *scaling functions*, owing to fact that the systematic manner in which our approximation space and the underlying mesh is refined allows for construction of bases for any V_n^p simply by scaling and translating the coarse mesh basis functions. For example, the basis functions for V_n^p can be chosen as

$$\phi_{k,j}^n = 2^{n/2} \phi_k(2^n(x + 1) - 2j - 1), \quad k = 0, \dots, p, \quad j = 0, \dots, 2^n - 1.$$

We can then express our global modal approximation on a mesh of $N = 2^n$ elements as

$$(2.1) \quad u_h^N(x) = 2^{-n/2} \sum_{j=0}^{2^n-1} \sum_{k=0}^p u_j^k \phi_{k,j}^n(x).$$

The intermediary information which is required to advance from a coarse approximation space V_n^p to a finer approximation space V_{n+1}^p is contained within the wavelet space W_n^p which is defined as the orthogonal compliment of V_n^p in V_{n+1}^p :

$$V_{n+1}^p = V_n^p \oplus W_n^p, \quad W_n^p \subset V_{n+1}^p, \quad W_n^p \perp V_n^p.$$

Inductively, this hierarchy allows the expression of the fine-mesh approximation space as the direct sum of the coarsest mesh approximation spaces and the intermediary wavelet spaces:

$$V_n^p = V_0^p \oplus W_0^p \oplus W_1^p \oplus \dots \oplus W_{n-1}^p.$$

Alternatively, this means our fine-mesh approximations are simply coarse mesh approximations plus details contained within these wavelet spaces. We denote the basis functions for W_n^p by $\psi_{k,j}^n$, $k = 0, \dots, p$, $j = 1, \dots, 2^n - 1$. These functions are known as multi-wavelets. Decomposing our approximation in an analogous manner to our approximation spaces, we have that our fine-mesh approximation is just the sum of a coarse mesh approximation (on a mesh consisting of N elements) and the details provided by the multi-wavelets:

$$u_h^f(x) = u_h^N(x) + \sum_{j=0}^{2^n-1} \sum_{k=0}^p d_{k,j}^n \psi_{k,j}^n(x).$$

An illustration of this idea in two-dimensions is given in Figure 2.1. As a result, we see that one manner of obtaining fine-mesh approximations is to take a coarse-mesh approximation and add a linear combination of multi-wavelets, where the detail coefficient $d_{k,j}^n$ serve as the weights. Note that the magnitude of the detail coefficients $d_{k,j}^n$ is determined by the local regularity of the function being approximated [7]. The difficulty is that when approximation information is only given on a coarse grid, we do not have these finer details. SIAC-MRA [17] enables a way of approximating detail coefficients $d_{k,j}^n$ when only coarse mesh information is given. Or, as an alternative for moving approximations from a coarse mesh to a fine mesh.

2.3. Multi-dimensional SIAC Filter. The Smoothness-Increasing Accuracy-Conserving (SIAC) post-processor is a convolution kernel originally designed to enable superconvergence in finite element methods [6, 20]. They have since been extended for derivatives, boundaries and nonlinear hyperbolic equations [18, 19, 12, 13, 23] as well as mesh adaptivity [8]. The SIAC filter has been generalized for application to higher dimensional data using two methods [10, 20]: the first is a tensor product SIAC filter, whereby a convolution kernel is constructed by taking the tensor product of multiple one-dimensional SIAC kernels; the second is to construct a Line SIAC filter (L-SIAC), which is simply a one-dimensional SIAC kernel that has been rotated to align with a non-Cartesian axis. Both of these techniques rely on understanding the one-dimensional kernel presented in this section.

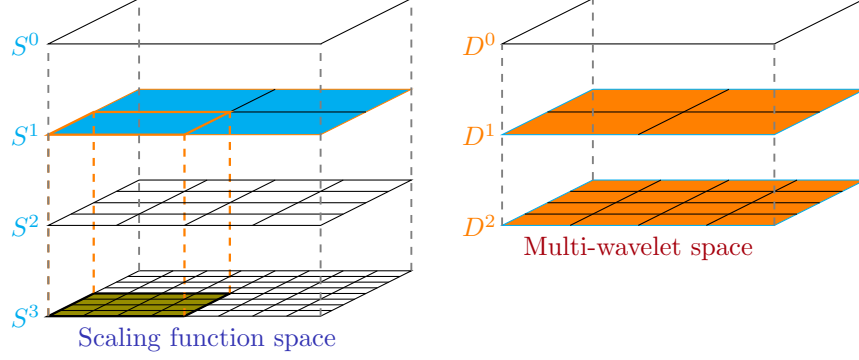


FIG. 2.1. Illustration of the two-dimensional multi-wavelet idea. The approximation at the finest level, S^3 , can be represented as the approximation from the S^1 scaling function space plus the multi-wavelet coefficients from wavelet spaces D^1 and D^2 .

In one dimension, the SIAC filtered approximation is defined by

$$(2.2) \quad u_h^*(x) = K_H \star u_h = \int_{-\infty}^{\infty} K_H^{(r+1, \ell)}(x-y) u_h(y) dy,$$

where the convolution kernel $K_H(x) = \frac{1}{H} K\left(\frac{x}{H}\right)$ is a scaled linear combination of $r+1$ central B-splines of order ℓ , that is normalized in L^1 :

$$K^{(r+1, \ell)}(t) = \sum_{\gamma=-r/2}^{r/2} c_\gamma B^{(\ell)}(t + \gamma).$$

The coefficients c_γ are chosen so that the kernel reproduces polynomials (consistency and moments) up to degree r :

$$K^{(r+1, \ell)} \star x^k = x^k \quad \text{for } k = 0, \dots, r,$$

while the central B-splines are piecewise polynomials defined recursively via the relation

$$\begin{aligned} B^{(1)} &= \chi_{[-1/2, 1/2)} \\ B^{(n+1)} &= B^{(n)} \star B^{(1)} \\ &= \frac{1}{n} \left[(n/2 + t) B^{(n)}(t + 1/2) + (n/2 - t) B^{(n)}(t - 1/2) \right]. \end{aligned}$$

Central B-splines are used in the kernel construction as they provide finite support to the kernel, while allowing derivatives of the kernel to be expressed as divided differences. This latter property and the preservation of moments condition are crucial in deriving error estimates [6, 12, 15]. An introduction to B-splines and their properties can be found in [5]. The kernel is scaled by a parameter H typically set as the uniform element width h . To enable superconvergence in DG approximations, we require $r = 2p$ and $\ell = p + 1$. This allows for our filtered approximation to obtain $\mathcal{O}(h^{2p+1})$ errors in the L^2 -norm and L^∞ -norm for linear hyperbolic equations.

In two dimensions, the Line SIAC filter is given by

$$(2.3) \quad u_h^*(\mathbf{x}) = \int_{\Gamma} K_{\Gamma, H}^{(r+1, \ell)}(t) u_h(t) dt,$$

where

$$\Gamma(t) = \mathbf{x} + t(\cos(\theta), \sin(\theta)),$$

and

$$(2.4) \quad K_{\Gamma, H}^{(r+1, \ell)}(t) = \sum_{\gamma=-r/2}^{r/2} c_\gamma B_H^{(\ell)}(t - \gamma) = \frac{1}{H} \sum_{\gamma=-r/2}^{r/2} c_\gamma B^{(\ell)}\left(\frac{t}{H} - \gamma\right).$$

The angle of rotation θ and the scaling parameter H are selected to be

$$(2.5) \quad \theta = \tan^{-1} \left(\frac{h_{x_2}}{h_{x_1}} \right), \quad H = h_{x_1} \cos(\theta) + h_{x_2} \sin(\theta).$$

In this article we choose the uniform mesh diagonal, $\theta = \pi/4$, and subsequently $H = \sqrt{2}h_{x_1} = \sqrt{2}h_{x_2}$. In higher dimensions we simply need to choose a new orientation and scaling factor. In three-dimensional space, the filtered approximation is given similarly where the line Γ parameterized by t is given by

$$(2.6) \quad \Gamma(t) = \mathbf{x} + t(\cos(\theta) \sin(\phi), \sin(\theta) \sin(\phi), \cos(\phi)) = \mathbf{x} + t\mathbf{v},$$

with θ, ϕ being the orientation of the line in spherical coordinates. Because our approximation lies on a uniform three-dimensional mesh, the line filter will be aligned with one of the four diagonals connecting antipodal points of the cubic element. Hence, we choose $H = \sqrt{3}h$, where h is the uniform mesh spacing, and $\theta = \pi/4$, $\phi = \tan^{-1}(\sqrt{2})$. This results in the direction vector $\mathbf{v} = \frac{1}{\sqrt{3}}(1, 1, 1)$.

The utility of the L-SIAC filter is its ability to filter higher-dimensional data without increasing the dimensionality of the kernel support. This makes it an alternative to the tensor product filter when evaluating the convolution via quadrature sums. For example, in 2D for a quadratic approximation it can reduce the number of quadrature sums from 4000 to ten [10]. Additionally, the constant in the error estimate is reduced for the filtered solution. We consider the L-SIAC filter specifically because of its reduced and rotated kernel support in this context. The ability to manipulate the kernel orientation and dimensionality is anticipated to be useful in generalizing to triangulations due to the nature of the multi-variate basis functions.

3. Multi-dimensional Line SIAC-MRA. In this section, we introduce Line SIAC for more accurate multi-resolution analysis. This procedure allows us to obtain detail coefficients for moving information to successively finer grids. *These detail coefficients are obtained for a uniform mesh by setting $r = 2p$ and $\ell = 1$ in the LSIAC filter* (Equation (2.4)). This ensures enough moments in the underlying data are respected as well as ensuring that over-smoothing does not occur. For this choice of parameters, the filter coefficients are given by the equation

$$(3.1) \quad \sum_{\gamma=-p}^p c_\gamma \frac{1}{m+1} \left[\left(\frac{1}{2} - \gamma \right)^{m+1} + \left(\frac{1}{2} + \gamma \right)^{m+1} \right] = \begin{cases} 1, & m = 0 \\ 0, & m = 1, \dots, 2p \end{cases}.$$

As presented in [17], the one-dimensional convolution, (2.2), can be analytically calculated and thereby allows for expressing our filtered approximation in a new basis. As a result of the ordering of mesh and kernel breaks, we have a piecewise definition for the filtered approximation itself:

$$u_h^*(x) = \begin{cases} \sum_{m=0}^{p+1} d_m^j \chi_L^m(\zeta), & \zeta \in (-1, 0) \\ \sum_{m=0}^{p+1} \hat{d}_m^j \chi_R^m(\zeta), & \zeta \in (0, 1) \end{cases},$$

for each element. Here $\zeta = \xi_j(x)$ represents a local element mapping. The filtered basis functions χ_L^m and χ_R^m used in this article and their associated coefficients are given in the supplementary materials. Assuming that our initial approximation is in V_n^p and letting N be the number of elements of this approximation space, denote the fine-mesh approximation by $u_h^f = \mathcal{P}_{n+1} u_h^*$, where \mathcal{P}_{n+1} represents the projection operator onto the approximation space V_{n+1}^p consisting of $2N$ elements. To obtain the fine-mesh modal coefficients in the ansatz $u_h^f(x)|_{I_n} = \sum_{k=0}^p u_k^n \phi_k^n(x)$, $n = 1, \dots, 2N$ we require

$$\langle u_h^f, \phi_m^j \rangle_{I_n} = \langle u_h^*, \phi_m^j \rangle_{I_n}, \quad m = 0, \dots, p.$$

Having determined our fine-mesh approximation $u_h^f = \mathcal{P}_{n+1} u_h^*$, we can isolate the multi-wavelet component simply by subtraction of the coarse-approximation in Equation (2.1):

$$(3.2) \quad \sum_{j=0}^{2^{n-1}-1} \sum_{k=0}^p \hat{d}_{k,j}^n \psi_{k,j}^n(x) = \mathcal{P}_{n+1} u_h^* - u_h,$$

where $\psi_{k,j}^n$ are the basis functions of the multi-wavelet space. These alternative multi-wavelet coefficients provide an improvement to our approximation, and the SIAC-MRA procedure provides a definition for them. This allows for effectively transitioning the approximation onto a finer mesh. Additionally, it only requires coarse mesh data to be constructed. This differs significantly from traditional MRA where multi-wavelet coefficients are intrinsically dependent upon the initial function u , which means that we cannot construct multi-wavelet coefficients for improving our approximation if only coarse modal data is available.

Knowing the form of the filtered approximation, we can map the coefficients to the fine-mesh modes simply by application of a projection operator. Furthermore, knowing the forms of the filtered approximation allows us to discretize this procedure and construct projection matrices to perform this scale transition. This will be discussed further in the implementation Section 4.

As demonstrated in [17], repeated application of the filtering-projection procedure results in increasingly better approximations. It should be noted that SIAC-MRA is mesh-dependent in that our ability to simplify the convolution is intrinsically linked to patterns in the mesh construction. In this paper we consider uniform quadrilateral and hexahedral meshes, to generalize this procedure to higher dimensional spaces and introduce LSIAC-MRA.

In a manner analogous to the one-dimensional case [17], we can take advantage of the simplicity of the kernel's B-spline components and the predictability of the element boundaries to analytically evaluate the convolutions given by Equation (2.3). Following this procedure, and letting $\zeta_x = \xi_i(x)$, $\zeta_y = \xi_j(y)$, and $\zeta_z = \xi_k(z)$ denote the mapping of our filtering point to the reference element, we obtain the following expressions for our filtered approximations:

For the two-dimensional Line SIAC, the filtered approximation can be written as

$$(3.3) \quad u_h^*(x, y) = \sum_{P=\{L,M,R\}} \sum_{m=0}^p \sum_{r=0}^p a_{m,r}^{S,P,i,j} \chi_{S,P}^{m,r}(\zeta_x, \zeta_y),$$

where (x, y) belongs to region S of element (i, j) . These regions are delineated in Figure 4.1. The P index results from splitting up the convolution into “left”, “middle”, and “right” components owing to element breaks. Analogously, the three-dimensional LSIAC filtered approximation can be written as

$$(3.4) \quad u_h^*(x, y, z) = \sum_{P=\{L,LM,RM,R\}} \sum_{m=0}^p \sum_{r=0}^p \sum_{\ell=0}^p a_{m,r,\ell}^{S,P,i,j,k} \chi_{S,P}^{m,r,\ell}(\zeta_x, \zeta_y, \zeta_z),$$

where (x, y, z) belongs to region S of element (i, j, k) , where again the P index denotes an ordering with respect to element breaks. A description of the aforementioned regions and an associated classification scheme are provided in the supplementary materials.

In two dimensions, the LSIAC implementation requires 6 quadrature domains per element. In three dimensions there are 24 quadrature domains per element. Note that for the L-SIAC filters, more complicated integrals of shifted Legendre Polynomials are produced. Though the reduced dimension of the L-SIAC kernel support means less information is needed to compute the a coefficients, the integrand is no longer separable which leads to more complicated functions in the expansion. All the χ functions and their associated coefficients are detailed in the supplementary materials. Much the same as in one dimension, we can project these filtered approximations onto the finer mesh, and thereby construct fine-mesh approximations. The only difference is that we are now projecting onto either the quadrant or octants of reference element in two and three dimensions respectively. In the L-SIAC case this requires us to account for the piecewise definition of the filtered approximations and split up projections according to the associated domains of definition.

In what follows, we demonstrate the performance of LSIAC-MRA. Because $\psi^{(1)}(x) = \chi_{[-1/2,1/2]}(x)$, we are able to analytically evaluate the convolution for the L-SIAC kernel by easily accounting for mesh and kernel breaks. This enables us to obtain a closed form expression for the filtered approximation expressed in a new basis and allows for our improved multi-resolution analysis technique.

4. Implementation. In this section we describe a brief overview of the implementation of 2D LSIAC-MRA as the same ideas extend to higher dimensions. Additional details on the filtered approximation basis functions and a computational algorithm are included in the supplementary materials.

4.1. Transition Operators. Given coarse-mesh modal information, we wish to construct transition matrices for mapping the coarse-mesh modal coefficients to their refined fine-mesh counterparts. In our

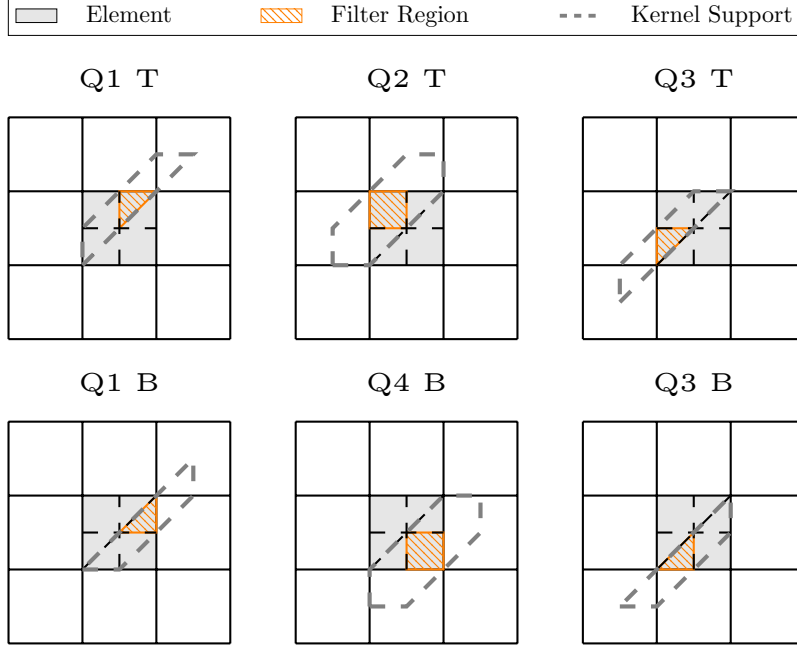


FIG. 4.1. 2D depiction of the varying element overlap of the L-SIAC kernel resulting in varying filtered approximation forms for $p = 0$. The element is shaded in gray, filter region has diagonal lines, and the kernel support is encapsulated by dashed lines. The filter region represents the location of possible filtering points, and the kernel support represents the overlap of the kernel support for those filtering points.

discrete setting these transition matrices are the composition of the discrete filtering and projection matrices $T = \mathcal{P}_{n+1}K_H$. In two dimensions we will have a separate transition matrix for each quadrant. It is important to note that the sequencing of mesh and kernel breaks, and the elements within the kernel's support varies from region to region (see Figure 4.1). As can be seen when computing the refined approximation, we must project our filtered approximation onto the finer mesh. In the 2D L-SIAC case we must sum the projections over the two triangular regions composing quadrants I and III. This requires different transition matrix construction from quadrant-to-quadrant. In the following we outline the construction procedure and refer the reader to Figure 4.2 for a depiction of the effects of each matrix in the composition for the case $p = 0$.

For the purposes of performing the enhancement procedure via matrix operations it is important to have a standard ordering for the modal information. As constructing the transition matrices is easier if modal coefficients are grouped by element, we denote the $(N^2(p+1))^2 \times 1$ vector of modal coefficients by

$$\vec{u} = [\vec{u}^1 \quad | \quad \dots \quad | \quad \vec{u}^N]^T,$$

where

$$\vec{u}^j = [\vec{u}^{1,j} \dots \vec{u}^{N,j}],$$

$$\vec{u}^{i,j} = [\vec{u}_0^{i,j} \quad | \quad \dots \quad | \quad \vec{u}_p^{i,j}],$$

and

$$\vec{u}_{ky}^{i,j} = [u_{0,ky}^{i,j} \quad | \quad \dots \quad | \quad u_{p,ky}^{i,j}].$$

4.1.1. Sifting Matrices. All of the coefficients in (3.3) fit the form

$$(4.1) \quad a_{m,r}^{i,j}(q_x, q_y) = \sum_{\gamma=-p}^p c_\gamma u_{kx,ky}^{i+\gamma+q_x, j+\gamma+q_y},$$

where $q_x, q_y \in \{-1, 0, 1\}$ are uniquely determined by the region S and position P as detailed in the supplementary materials. To compute these coefficients, a selection matrix, denoted $\mathcal{S}^{i,j}$, is first constructed.

Assuming periodic boundary conditions, this matrix will select all modal values from the $(2p+3)^2$ elements about element (i, j) . We will later further pair this information to only those elements under the line filter's support. Define the $(2p+3)N(p+1)^2 \times N^2(p+1)^2$ matrix S^j by

$$S^j = \begin{bmatrix} \ddots & & \vdots & & \ddots \\ \dots & \omega_j(m, r) \mathbb{1}_{N(p+1)^2 \times N(p+1)^2} & \dots & & \dots \\ \ddots & & \vdots & & \ddots \end{bmatrix}$$

where

$$\omega_j(m, r) = \begin{cases} 1, & j \neq N \ \& \ p+2-m+r = j \pmod{N} \\ 1, & j = N \ \& \ p+2-m+r = j \text{ or } 0 \\ 0, & \text{else} \end{cases},$$

for $m = 1 : (2p+3)$ and $r = 1 : N$. This matrix selects the relevant j -coordinates:

$$S^j \vec{u} = [u^{j-(p+1)} \dots u^{j+(p+1)}]^T.$$

Next, construct an $(2p+3)(p+1)^2 \times N(p+1)^2$ matrix \tilde{S}^i to select the relevant i -coordinates from within each vector. Define

$$\tilde{S}^i = \begin{bmatrix} \ddots & & \vdots & & \ddots \\ \dots & \omega_i(m, r) \mathbb{1}_{(p+1)^2 \times (p+1)^2} & \dots & & \dots \\ \ddots & & \vdots & & \ddots \end{bmatrix}$$

for $m = 1 : (2p+3)$ and $r = 1 : N$. We have that

$$\tilde{S}^i u^j = [u^{i-(p+1),j} \dots u^{i+(p+1),j}]^T.$$

Hence, defining $\mathcal{S}^{i,j}$ to be the $(2p+3)^2(p+1)^2 \times N^2(p+1)^2$ matrix given by

$$\mathcal{S}^{i,j} = \begin{bmatrix} \tilde{S}^i & & 0 \\ & \ddots & \\ 0 & & \tilde{S}^i \end{bmatrix} S^j,$$

where there are $(2p+3)$ \tilde{S}^i matrices on the diagonal, we have

$$\mathcal{S}^{i,j} \mathbf{u} = \left[[u^{i-(p+1),j-(p+1)} \dots u^{i+(p+1),j-(p+1)}] \dots [u^{i-(p+1),j+(p+1)} \dots u^{i+(p+1),j+(p+1)}] \right]^T = \tilde{u}^{i,j}$$

which, as stated previously, only contains modal information from the $(2p+3)^2$ elements around element (i, j) .

Now we construct an intermediary matrix \mathcal{C}^{q_x, q_y} whose function is to select the elements from $\tilde{u}^{i,j}$ which fall within the support of the line integral as affected by the shifting arguments (q_x, q_y) . Define the $(2p+1)(2p+3)(p+1)^2 \times (2p+3)^2(p+1)^2$ matrix C^{q_y} by

$$C^{q_y} = \begin{bmatrix} \delta_{-1, q_y} \mathbb{1} & \delta_{0, q_y} \mathbb{1} & \delta_{1, q_y} \mathbb{1} & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \delta_{-1, q_y} \mathbb{1} & \delta_{0, q_y} \mathbb{1} & \delta_{1, q_y} \mathbb{1} \end{bmatrix},$$

where δ is the Kronecker delta and $\mathbb{1}$ is the $(2p+3)(p+1)^2 \times (2p+3)(p+1)^2$ identity matrix. This matrix has the effect of further restricting the relevant j -indices based off the shifting argument q_y . Then define the $(2p+1)(p+1)^2 \times (2p+3)(2p+1)(p+1)^2$ matrix \mathcal{C}^{q_x} by

$$\mathcal{C}^{q_x} = \begin{bmatrix} \delta_{-1, q_x} \mathbb{1} & \delta_{0, q_x} \mathbb{1} & \delta_{1, q_x} \mathbb{1} & 0 \\ & & R_2 & \\ & & \vdots & \\ & & R_{2p+1} & \end{bmatrix},$$

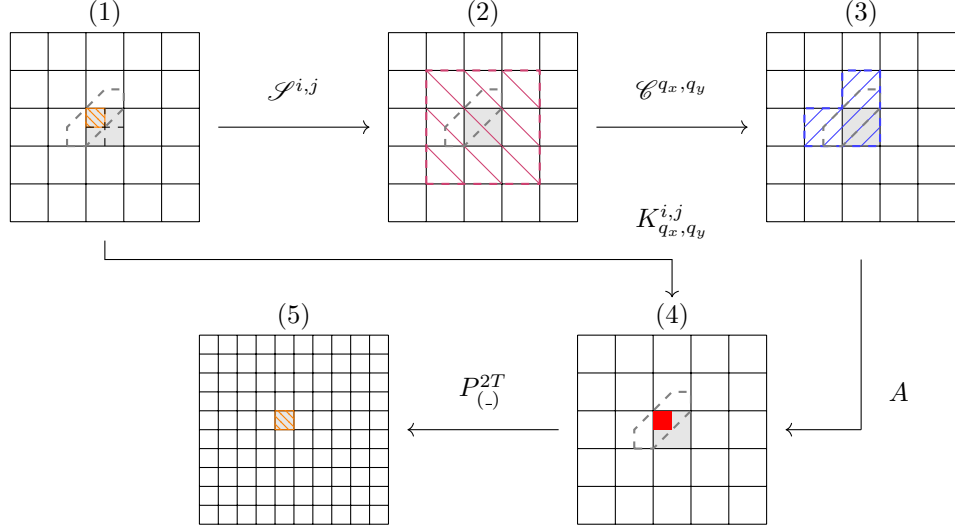


FIG. 4.2. Purpose of transition matrix components during refinement procedure: (1) \rightarrow (2) Select modes from elements relevant to refining element (i, j) , (2) \rightarrow (3) Select only the modes relevant to filtering a given region, in this case $2T$, (3) \rightarrow (4) Use that modal information to obtain filtered approximation coefficients, (4) \rightarrow (5) Obtain fine mesh modes by projecting the filtered approximation onto the refined mesh. Notice that $K_{q_x, q_y}^{i, j} = A \mathcal{C}^{q_x, q_y} \mathcal{S}^{i, j}$.

where $\mathbb{1}$ is $(p+1)^2 \times (p+1)^2$, and R_j is the first block row defined above with the non-zero entries translated $(j-1)(2p+4)(p+1)^2$ places to the right. We then have the desired $(2p+1)(p+1)^2 \times (2p+3)^2(p+1)^2$ intermediary matrix \mathcal{C}^{q_x, q_y} given by the composition

$$\mathcal{C}^{q_x, q_y} = \mathcal{C}^{q_x} C^{q_y}.$$

Applying this matrix produces:

$$\mathcal{C}^{q_x, q_y} \tilde{u}^{i, j} = \tilde{u}_{q_x, q_y}^{i, j} = [u^{i-(p-q_x), j-(p-q_y)}, \dots, u^{i+(p+q_x), j+(p+q_y)}]^T,$$

which are the modal coefficients appearing in (4.1).

4.1.2. Coefficient Matrices. We now describe the construction of a matrix for computing (4.1) from the sifted modal coefficients. Define the matrix A to be the $(p+1)^2 \times (2p+1)(p+1)^2$ matrix given by

$$A = \frac{1}{2} \begin{bmatrix} c_{-p} \mathbb{1} & & 0 & | & \dots & | & c_p \mathbb{1} & & 0 \\ 0 & c_{-p} \mathbb{1} & & | & \dots & | & 0 & c_p \mathbb{1} & 0 \\ & & \ddots & | & \dots & | & & \ddots & \\ 0 & & & c_{-p} \mathbb{1} & | & \dots & | & 0 & c_p \mathbb{1} \end{bmatrix}$$

where the identity matrices are $(p+1) \times (p+1)$. This is the matrix used to perform the summation and leads to

$$A \tilde{u}_{q_x, q_y}^{i, j} = \bar{a}^{i, j}(q_x, q_y)$$

where

$$\bar{a}^{i, j}(q_x, q_y) = [a_{r=0}^{i, j} \dots a_{r=p}^{i, j}]^T(q_x, q_y)$$

with

$$a_r^{i, j}(q_x, q_y) = [a_{0, r}^{i, j}(q_x, q_y) \dots a_{p, r}^{i, j}(q_x, q_y)].$$

Hence, the seven coefficient matrices $K_{q_x, q_y}^{i, j}$ are defined by the composition

$$K_{q_x, q_y}^{i, j} = A \mathcal{C}^{q_x, q_y} \mathcal{S}^{i, j}$$

and give

$$K_{q_x, q_y}^{i,j} u = \vec{a}^{i,j}(q_x, q_y).$$

Rather than concatenating these matrices at this point in the procedure, we will wait until after applying the projection in order to minimize the number of transformations performed. It is important to note that each choice of (q_x, q_y) computes a different set of coefficients pertaining to a different set of basis functions, each of which is required for obtaining the enhanced approximation.

4.1.3. Projection Matrices. We now discuss the procedure necessary to perform the projections. Because of the different filtered approximation forms in each quadrant, we must perform each projection slightly differently. Quadrants II and IV are similar as are I and III. Here we detail a procedure for QII.

QII. In QII we define the projection of this approximation by $u_h^f|_{K^{2i-1,2j}}(x, y)$ where the new modal coefficients are computed by

$$(4.2) \quad \langle u_h^*, \phi_{kx} \phi_{ky} \rangle_{QII} = \langle u_h^f, \phi_{kx} \phi_{ky} \rangle_{QII}$$

$$(4.3) \quad = u_{kx, ky}^{2i-1, 2j}, \quad kx, ky = 0, \dots, p, \quad i, j = 1, \dots, N.$$

To account for the change in domain scaling define $\xi_x(\zeta_x) = \frac{\zeta_x - 1}{2}$ and $\xi_y(\zeta_y) = \frac{\zeta_y + 1}{2}$. We have

$$u_{kx, ky}^{2i-1, 2j} = \sum_{m=0}^p \sum_{r=0}^p \int_{-1}^1 \int_{-1}^1 \left\{ a_{m,r}^{i,j}(-1, 0) \chi_{2T,L}^{m,r}(\xi_x, \xi_y) \right. \\ \left. + a_{m,r}^{i,j}(0, 0) \chi_{2T,M}^{m,r}(\xi_x, \xi_y) \right. \\ \left. + a_{m,r}^{i,j}(0, 1) \chi_{2T,R}^{m,r}(\xi_x, \xi_y) \right\} \phi_{kx}(\zeta_x) \phi_{ky}(\zeta_y) d\zeta_y d\zeta_x.$$

This expression will be broken up into three separate integrals, one for each of the different $\chi_{(\cdot)}$ functions where $(\cdot) \in \{L, M, R\}$. For each $\chi_{(\cdot)}$, define

$$P_{(\cdot)}(kx, ky, m, r) = \int_{-1}^1 \int_{-1}^1 \chi_{(2T, \cdot)}^{m,r}(\xi_x, \xi_y) \phi_{kx}(\zeta_x) \phi_{ky}(\zeta_y) d\zeta_y d\zeta_x.$$

Define the projection matrix $P_{(\cdot)}^{2T}$ by

$$P_{(\cdot)}^{2T} = [\vec{P}(0, 0) \quad \dots \quad \vec{P}(p, 0) \quad \dots \quad \vec{P}(0, p) \quad \dots \quad \vec{P}(p, p)]^T,$$

where

$$\vec{P}(kx, ky) = [P(kx, ky, 0, 0) \dots P(kx, ky, p, 0) \dots \dots P(kx, ky, 0, p) \dots P(kx, ky, p, p)]^T.$$

Hence, we have

$$u_{2i-1, 2j}^f = P_L^{2T} \vec{a}^{i,j}(-1, 0) + P_M^{2T} \vec{a}^{i,j}(0, 0) + P_R^{2T} \vec{a}^{i,j}(0, 1).$$

Noting the necessary choice of $K_{q_x, q_y}^{i,j}$ to obtain each of the coefficient vectors, we have

$$u_{2i-1, 2j}^f = (P_L^{2T} K_{-1,0}^{i,j} + P_M^{2T} K_{0,0}^{i,j} + P_R^{2T} K_{0,1}^{i,j}) \vec{u}.$$

Denoting this composition by $T_{i,j}^2$ we can concatenate these matrices together to obtain the global transition matrix T^2 defined by

$$T^2 = [T_{1,1}^2 \quad \dots \quad T_{N,1}^2 \quad | \dots | \quad T_{1,N}^2 \quad \dots \quad T_{N,N}^2]^T,$$

such that

$$u_2^f = T^2 u.$$

QIV. The projection matrices for quadrant IV are the same except that the quadrant IV basis function are used and the change of variables are instead $\xi_x(\zeta_x) = \frac{\zeta_x + 1}{2}$ and $\xi_y(\zeta_y) = \frac{\zeta_y - 1}{2}$.

QIII. The cases of quadrants I and III are similar in that a change of variables must be applied to split the integrals over the quadrant into a pair of integrals over diagonals of that quadrant. Beginning with the integral over the whole quadrant, we have

$$\begin{aligned}\bar{u}_{kx,ky}^{2i-1,2j-1} &= \langle u_h^*, \phi_{kx}^i \phi_{ky}^j \rangle_{QIII} \\ &= \int_{-1}^1 \int_{-1}^1 u_h^* |_{QIII}(\xi_x(\zeta_x), \xi_y(\zeta_y)) \phi_{kx}^i(\zeta_x) \phi_{ky}^j(\zeta_y) d\zeta_y d\zeta_x,\end{aligned}$$

where $\xi_x = \frac{\zeta_x - 1}{2}$ and $\xi_y = \frac{\zeta_y - 1}{2}$. Breaking this up into integrals over $3T$ and $3B$ we have

$$\begin{aligned}\bar{u}_{kx,ky}^{2i-1,2j-1} &= \int_{-1}^1 \int_{\zeta_x}^1 u_{3T}^*(\xi_x, \xi_y) \phi_{kx}^i(\zeta_x) \phi_{ky}^j(\zeta_y) d\zeta_y d\zeta_x \\ &\quad + \int_{-1}^1 \int_{-1}^{\zeta_x} u_{3B}^*(\xi_x, \xi_y) \phi_{kx}^i(\zeta_x) \phi_{ky}^j(\zeta_y) d\zeta_y d\zeta_x.\end{aligned}$$

Now to use Gauss-Legendre quadrature, we must scale the bounds of the inner integral to $[-1, 1]$. To do so introduce the change of variables $\zeta_y = \frac{1-\zeta_x}{2}\alpha_T + \frac{1+\zeta_x}{2}$ for the first integral and $\zeta_y = \frac{\zeta_x-1}{2}\alpha_B + \frac{1+\zeta_x}{2}$ for the second. This allows us to rewrite the expression as

$$\begin{aligned}\bar{u}_{kx,ky}^{2i-1,2j-1} &= \int_{-1}^1 \int_{-1}^1 u_{3T}^*(\xi_x, \xi_y(\zeta_y(\alpha_T))) \phi_{kx}^i(\zeta_x) \phi_{ky}^j\left(\frac{1}{2}[\alpha_T(1-\zeta_x) + 1 + \zeta_x]\right) \frac{1-\zeta_x}{2} d\alpha_T d\zeta_x \\ &\quad + \int_{-1}^1 \int_{-1}^1 u_{3B}^*(\xi_x, \xi_y(\zeta_y(\alpha_B))) \phi_{kx}^i(\zeta_x) \phi_{ky}^j\left(\frac{1}{2}[\alpha_B(\zeta_x-1) + 1 + \zeta_x]\right) \frac{\zeta_x-1}{2} d\alpha_B d\zeta_x. \\ &= \mathcal{I}^T + \mathcal{I}^B.\end{aligned}$$

We now need to split the \mathcal{I} terms into a sum of integrals over the basis functions and then multiply by the appropriate modal coefficients. Defining

$$P_{(-)}^{3T}(kx, ky, m, r) = \int_{-1}^1 \int_{-1}^1 \chi_{3T,(-)}^{m,r}(\xi_x, \xi_y(\alpha_T)) \phi_{kx}(\zeta_x) \phi_{ky}(\zeta_y(\alpha_T)) d\zeta_x d\alpha_T,$$

where $(-) \in \{L, M, R\}$, we construct the projection matrices $P_L^{3T}, P_M^{3T}, P_R^{3T}$ in the same manner as in quadrant II. Now for the second set of integrals define

$$P_{(-)}^{3B}(kx, ky, m, r) = \int_{-1}^1 \int_{-1}^1 \chi_{3B,(-)}^{kx,ky}(\xi_x, \xi_y(\alpha_B)) \phi_{kx}(\zeta_x) \phi_{ky}(\zeta_y(\alpha_B)) d\zeta_x d\alpha_B,$$

and analogously construct the projection matrices $P_L^{3B}, P_M^{3B}, P_R^{3B}$. We then have

$$\begin{aligned}u_{2i-1,2j-1}^f &= P_L^{3T} \bar{a}^{i,j}(-1, -1) + P_M^{3T} \bar{a}^{i,j}(-1, 0) + P_R^{3T} \bar{a}^{i,j}(0, 0) \\ &\quad + P_L^{3B} \bar{a}^{i,j}(-1, -1) + P_M^{3B} \bar{a}^{i,j}(0, -1) + P_R^{3B} \bar{a}^{i,j}(0, 0).\end{aligned}$$

Now $\bar{a}^{i,j}(q_x, q_y) = K_{q_x, q_y}^{i,j} \bar{u}$, thus

$$u_{2i-1,2j-1}^f = \left[(P_L^{3T} + P_L^{3B}) K_{-1,-1}^{i,j} + (P_R^{3T} + P_R^{3B}) K_{0,0}^{i,j} + P_M^{3T} K_{-1,0}^{i,j} + P_M^{3B} K_{0,-1}^{i,j} \right] \bar{u}.$$

Denoting this composition by $T_{i,j}^3$ we can concatenate these matrices together to obtain the global transition matrix T^3 defined by

$$T^3 = [T_{1,1}^3 \quad \dots \quad T_{N,1}^3 \quad | \dots | \quad T_{1,N}^3 \quad \dots \quad T_{N,N}^3]^T,$$

such that

$$u_3^f = T^3 u.$$

QI. The case of quadrant one is that same as quadrant three except now $\xi_x = \frac{\zeta_x + 1}{2}$, $\xi_y = \frac{\zeta_y + 1}{2}$, and the χ functions used are those defined on quadrant one.

Coding Comment: The LSIAC-MRA procedure was implemented in MATLABTM R2021a. The authors make no claim to the computational efficiency of the implementation but simply state that the memory requirements are on the order of the number of modal values of the finer mesh $4N^2(p+1)^2$. An algorithm, whereby only the local transition matrices $T_{i,j}^Q$ are formed to perform the enhancement is provided in the supplementary materials.

5. Numerical Results. The goal of LSIAC-MRA is to produce a fine-mesh modal approximation from coarse-mesh data that has lower error than repeated L^2 -projection alone. The measures used for comparison are the L^2 - and L^∞ -errors defined respectively by

$$(5.1) \quad \|u_{exact} - u_{approx}\|_0 = \sqrt{\frac{1}{|\Omega|} \int_{\Omega} |u_{exact} - u_{approx}|^2 d\Omega}, \quad \|u_{exact} - u_{approx}\|_\infty = \sup_{x \in \Omega} |u_{exact} - u_{approx}|.$$

For our simulations, the L^2 error is evaluated through Gauss-Legendre quadrature at 6^d nodes per element. Similarly, the L^∞ is taken to be the maximum absolute error over these nodes. To ensure a consistent standard for comparison, we compute the errors on the finest mesh occurring during the enhancement procedure. It is very important to use a standard mesh for computing the errors. Not doing so can lead to counter-intuitive behaviors in the error. An example of this is displayed in Figure 5.2 where repeated projections of the same polynomial approximation have varying errors, contradicting the reproduction property of the projection. The cause is the discrete approximation of the error-norms. By performing the error computation over a fixed sufficiently fine mesh, we no longer encounter these discretization artifacts. In all the simulations considered below, periodic boundary conditions are assumed to allow the application of the filter in regions where the kernel support would surpass the domain boundaries. The first two columns of the tables detail the projections errors of the L^2 approximation subject to scale transition by projection without filtering. This represents the standard which our filtered approximations must outperform to justify the computational expense. The third and fourth columns detail the errors of an approximation when LSIAC-MRA is applied only once and then L^2 -projection is used. The final two columns detail the errors of the LSIAC-MRA procedure applied at each refinement. As a result, the first rows are equal. Similarly, the Enhanced Once and Enhanced Each Refinement columns will agree on the second mesh simply because only one filtering procedure has been applied at that point. These equalities are emphasized by the shading of entries in the tables (gray and medium gray, respectively). We have performed numerous two- and three-dimensional experiments to gauge the ability of the LSIAC-MRA filtering-projection procedure, but below we will only focus on the most illuminating examples. Note that we only expect reduced errors, not improvement in order of convergence.

5.1. 2D Test Problems. Consider the following functions on the domain $\Omega = \{(x, y) \in [0, 1]^2\}$:

1. $u_0(x, y) = \sin(2\pi(x + y))$
2. $u_0(x, y) = \sin(10\pi x) \sin(10\pi y)$
3. $u_0(x, y) = f_1(x)f_1(y)$
4. $u_0(x, y) = f_2(x)f_2(y)$

where

$$f_1(x) = \begin{cases} 2 \cos(2\pi(2x - 1)), & 1/4 \leq x \leq 3/4 \\ \cos(4\pi(2x - 1)), & Else \end{cases},$$

$$f_2(x) = \begin{cases} 2/3 \sin(2\pi(2x - 1)), & 1/4 \leq x \leq 3/4 \\ \cos(\pi(2x - 1)), & Else \end{cases}.$$

These tests aid us in evaluating the effectiveness of the LSIAC-MRA procedure for different function types. The first function can be viewed as a combination of a product and summation of the coordinate directions. The second represents a higher frequency function. The third is discontinuous, while the fourth has discontinuous derivatives for x and y at $1/4$ and $3/4$. Depictions of the third and fourth functions are provided in Figure 5.1.

Performing the LSIAC-MRA on functions with different wave number k , we observe that different initial mesh resolutions are necessary for improvement. The first and second functions behave similarly, except

that the first corresponds to a wavenumber $k = 1$ and has repeated improvement from LSIAC-MRA on a relatively coarse mesh (35×35 , see Figure 5.3). This contrasts with the higher frequency second function $k = 5$, which requires a much finer starting resolution of 160×160 for improvement. Even then, the resolution requirements increase with approximation order and so for coarser meshes the LSIAC-MRA procedure is more effective for lower order polynomial approximations, see Figure 5.5 and Table 5.1.

For the third and fourth functions we begin with their projection onto a piecewise orthonormal Legendre basis defined over a 70×70 uniform quadrilateral mesh. We observe that the refinement procedure provides error reduction for lower polynomial degrees, though performance degrades for increasing p . While LSIAC-MRA is effective in smooth regions, it is not as globally effective at error reduction as compared to the analytic examples above. This is due to the lack of smoothness of the functions under consideration. In cases 3 and 4, application of the SIAC filter near a discontinuity will smear the discontinuity by introducing artificial smoothness, and the pollution region will increase with each application of the filter. However, for these functions with non-smooth components occurring along lines, the pollution region will reach a finite extent of $h(2p+3/2)$ in any Cartesian direction from the point of discontinuity, where h is the uniform coarse mesh size. If we look at the contour plot given in Figure 5.4 or the zoomed in plots of Figure 5.7, we can in fact observe that near the discontinuous derivative of the function at $x, y = 1/4, 3/4$, increased errors occur with refinement. This error growth is lesser than compared to the discontinuous test case 3. We do not include refinement contour plots for test case 3 as the error distribution is similar to test case 4. Away from the polluted regions, significant error reduction occurs with refinement. The L^2 - and L^∞ -errors of the enhancement procedure *excluding the polluted regions* for these less smooth functions are given in Tables 5.2 and 5.3. We do not include the polluted errors because, as expected, the L^∞ -errors stay relatively constant while L^2 -errors grow as the polluted region covers a larger portion of the domain. The pollution regions for tests 3 and 4 are the same, and so we use an identical procedure for determining the pollution regions and computing the errors. The procedure is relatively simple, if an element on the coarse mesh contains a non-smooth feature caused by lack of smoothness in the initial condition, we exclude that element. When we enhance, we wish to exclude any element where the filtered approximation on that element drew information from polluted regions. As the non-smooth components of these functions occur along the lines $x, y = 1/4, 3/4$ we can exclude any element within $h(p+1/2)$ in any Cartesian direction of the non-smooth coarse elements on the refined mesh after application of LSIAC-MRA. Finally, following the second and final application of LSIAC-MRA, we exclude any element on the finest mesh within $\frac{3}{2}h(p+1/2)$ in any Cartesian direction of the non-smooth coarse elements. We observe that application of the filtering procedure at each stage continues to provide error reductions with each refinement and outperforms only applying the filtering procedure once, except in the $p = 4$ case for the third function. This is caused by insufficient resolution in the initial mesh, and increasing the initial resolution enables continued error reduction.

As depicted in the log-log plots in Figures 5.5 and 5.6, LSIAC-MRA can indeed provide error reduction for sufficiently smooth functions; however, this technique does not guaranteed an improvement in the order of the method. The contour plots in Figures 5.3 and 5.4 display that repeated application of LSIAC-MRA will alter not only the magnitude of errors, but also their distribution throughout the domain.

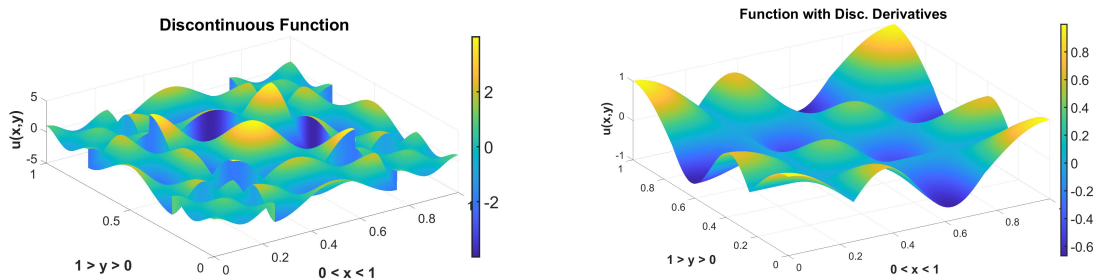


FIG. 5.1. Surface plots of $u_0(x, y) = f_1(x)f_1(y)$ (Left) and $u_0(x, y) = f_2(x)f_2(y)$ (Right). The initial condition defined as a tensor product of a discontinuous function f_1 represents a “strong discontinuity”, while that using a tensor product of a continuous but not globally differentiable function f_2 represents a continuous function with discontinuous derivatives $\frac{\partial u_0}{\partial x}$ and $\frac{\partial u_0}{\partial y}$.

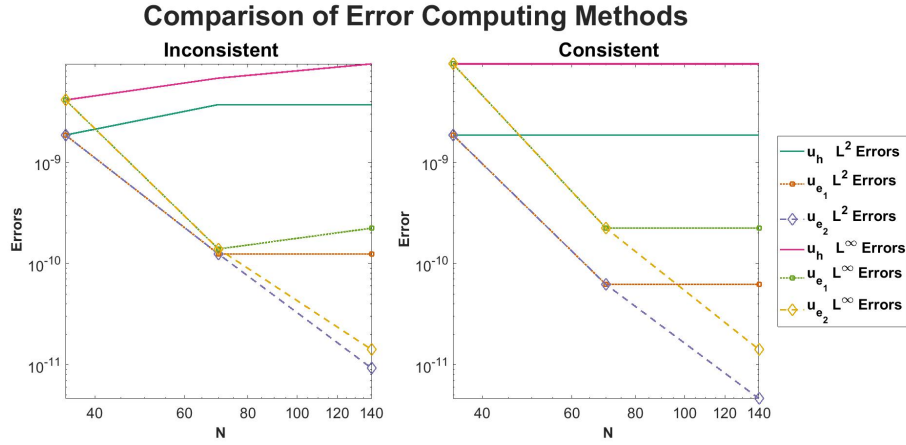


FIG. 5.2. Comparison of the log-log plots of the L^2 and L^∞ errors of the function $u_0(x, y) = \sin(2\pi x) + \sin(2\pi y)$ with $p = 4$ under refinement. In the left plot, the errors are computed on the mesh level of the iteration, i.e on the mesh size indicated by the data points abscissa. In the right plot, the errors are computed on the 140×140 mesh which is the finest mesh encountered during the procedure. We observe that this latter approach allows for the errors to remain constant under L^2 -projection alone, which agrees with the intuition of the approximation being reproduced.

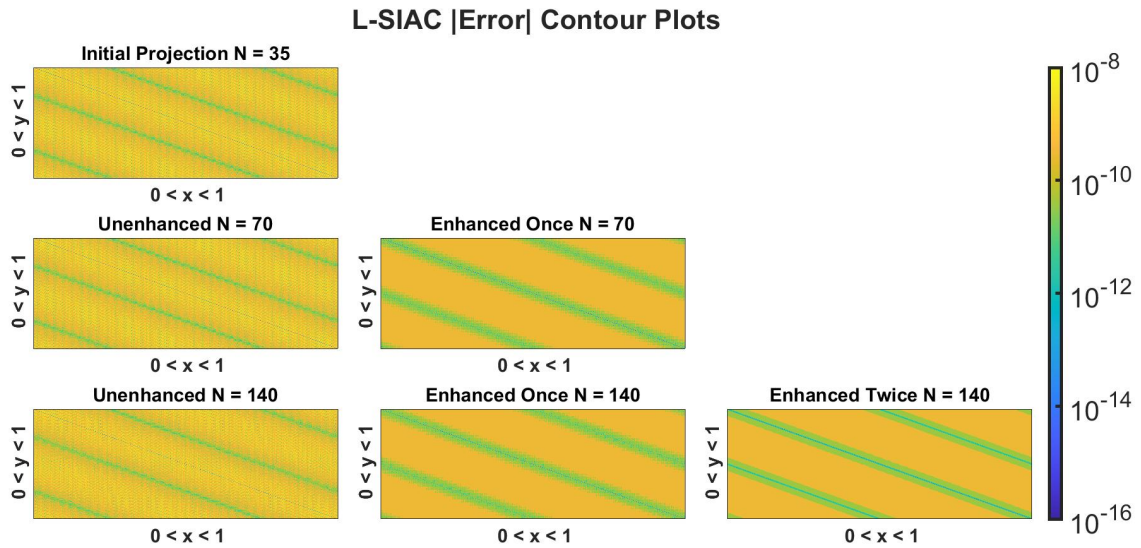


FIG. 5.3. Contour plots depicting pointwise $|Error|$ for initial condition $u_0(x, y) = \sin(2\pi(x + y))$ on a series of three meshes using $2D$ LSIAC-MRA for $p = 4$. Notice that the filtering procedure regularizes and reduces the errors when it is applied at each refinement.

L-SIAC |Error| Contour Plots

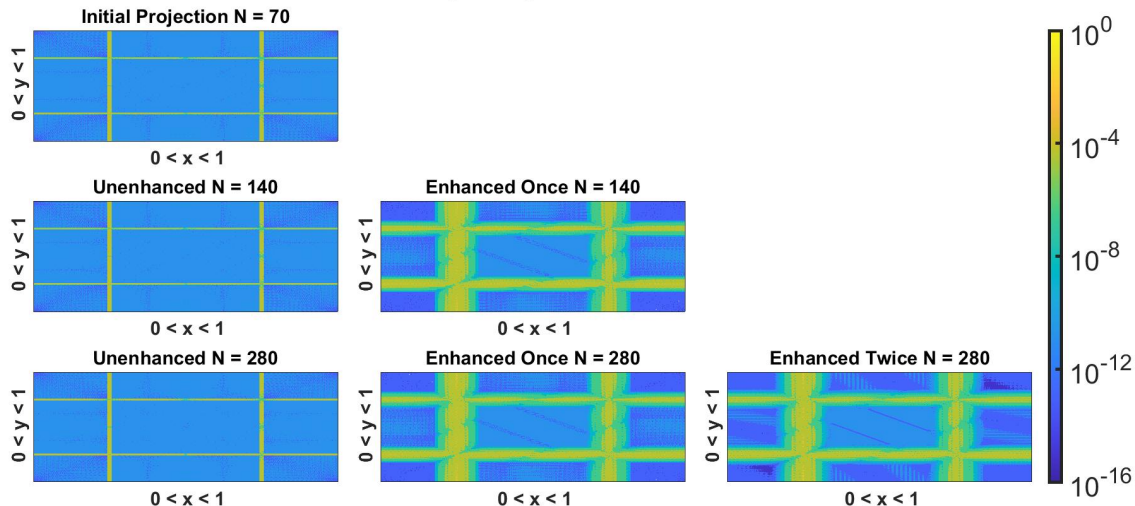


FIG. 5.4. Contour plots depicting pointwise $|\text{Error}|$ for the initial condition $u_0(x,y) = f_2(x)f_2(y)$ which contains a discontinuity in its first derivatives derivatives $\frac{\partial u_0}{\partial x}$ and $\frac{\partial u_0}{\partial y}$. 2D LSIAC-MRA was performed over a series of three meshes for $p = 4$. Notice that owing to the jump discontinuity in the derivatives at $x,y = 1/4, 3/4$, the polluted region where filtering adds artificial smoothness grows with each enhancement. Away from these locations, we still have error improvement. A zoomed in view is given in Figure 5.7.

TABLE 5.1

Table of L^2 and L^∞ errors for the high frequency initial condition $u(x,y) = \sin(10\pi x)\sin(10\pi y)$ on a series of three meshes using 2D LSIAC-MRA. Because it is high frequency, we have to initialize our approximation on a more refined grid. The shaded regions of the same shade represent the same approximation. For higher wave numbers the enhancement procedure requires greater initial mesh resolution for error reduction.

N	Projection Error		Enhanced Once		Enhanced Each Refinement	
	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error
\mathbb{P}^0						
160	$4.01e-02$	$9.59e-02$	$4.01e-02$	$9.59e-02$	$4.01e-02$	$9.59e-02$
320	$4.01e-02$	$9.59e-02$	$2.04e-02$	$5.30e-02$	$2.04e-02$	$5.30e-02$
640	$4.01e-02$	$9.59e-02$	$2.04e-02$	$5.30e-02$	$1.11e-02$	$2.97e-02$
\mathbb{P}^1						
160	$1.02e-03$	$6.10e-03$	$1.02e-03$	$6.10e-03$	$1.02e-03$	$6.10e-03$
320	$1.02e-03$	$6.10e-03$	$2.60e-04$	$1.40e-03$	$2.60e-04$	$1.40e-03$
640	$1.02e-03$	$6.10e-03$	$2.60e-04$	$1.40e-03$	$8.11e-05$	$3.08e-04$
\mathbb{P}^2						
160	$1.69e-05$	$6.07e-05$	$1.69e-05$	$6.07e-05$	$1.69e-05$	$6.07e-05$
320	$1.69e-05$	$6.07e-05$	$2.33e-06$	$9.44e-06$	$2.33e-06$	$9.44e-06$
640	$1.69e-05$	$6.07e-05$	$2.33e-06$	$9.44e-06$	$1.00e-06$	$2.97e-06$
\mathbb{P}^3						
160	$2.08e-07$	$1.48e-06$	$2.08e-07$	$1.48e-06$	$2.08e-07$	$1.48e-06$
320	$2.08e-07$	$1.48e-06$	$2.69e-08$	$1.08e-07$	$2.69e-08$	$1.08e-07$
640	$2.08e-07$	$1.48e-06$	$2.69e-08$	$1.08e-07$	$2.32e-08$	$3.63e-08$
\mathbb{P}^4						
160	$2.06e-09$	$7.91e-09$	$2.06e-09$	$7.91e-09$	$2.06e-09$	$7.91e-09$
320	$2.06e-09$	$7.91e-09$	$6.51e-10$	$1.19e-09$	$6.51e-10$	$1.19e-09$
640	$2.06e-09$	$7.91e-09$	$6.51e-10$	$1.19e-09$	$6.47e-10$	$9.39e-10$

TABLE 5.2

Table of L^2 and L^∞ errors for an initial condition that has discontinuities along $x, y = 1/4, 3/4$, $u(x, y) = f_1(x)f_1(y)$. The 2D LSIAC-MRA procedure was performed on a series of three meshes. Errors were calculated by excluding the polluted regions around the discontinuities. The LSIAC-MRA procedure reduces errors in these unpolluted regions.

N	Projection Error		Enhanced Once		Enhanced Each Refinement	
	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error
	\mathbb{P}^0					
70	$1.12e-01$	$3.52e-01$	$1.12e-01$	$3.52e-01$	$1.12e-01$	$3.52e-01$
140	$1.12e-01$	$3.52e-01$	$5.58e-02$	$1.92e-01$	$5.58e-02$	$1.92e-01$
280	$1.12e-01$	$3.52e-01$	$5.58e-02$	$1.92e-01$	$3.05e-02$	$1.07e-01$
	\mathbb{P}^1					
70	$4.00e-03$	$2.55e-02$	$4.00e-03$	$2.55e-02$	$4.00e-03$	$2.55e-02$
140	$4.00e-03$	$2.55e-02$	$9.08e-04$	$5.71e-03$	$9.08e-04$	$5.71e-03$
280	$4.00e-03$	$2.55e-02$	$9.08e-04$	$5.71e-03$	$3.55e-04$	$1.35e-03$
	\mathbb{P}^2					
70	$1.15e-04$	$6.95e-04$	$1.15e-04$	$6.95e-04$	$1.15e-04$	$6.95e-04$
140	$1.15e-04$	$6.95e-04$	$1.94e-05$	$1.04e-04$	$1.94e-05$	$1.04e-04$
280	$1.15e-04$	$6.95e-04$	$1.94e-05$	$1.04e-04$	$1.38e-05$	$6.15e-05$
	\mathbb{P}^3					
70	$2.47e-06$	$1.76e-05$	$2.47e-06$	$1.76e-05$	$2.47e-06$	$1.76e-05$
140	$2.47e-06$	$1.76e-05$	$1.09e-06$	$4.63e-06$	$1.09e-06$	$4.63e-06$
280	$2.47e-06$	$1.76e-05$	$1.09e-06$	$4.63e-06$	$9.30e-07$	$3.86e-06$
	\mathbb{P}^4					
70	$4.57e-08$	$3.00e-07$	$4.57e-08$	$3.00e-07$	$4.57e-08$	$3.00e-07$
140	$4.57e-08$	$3.00e-07$	$8.91e-08$	$3.55e-07$	$8.91e-08$	$3.55e-07$
280	$4.57e-08$	$3.00e-07$	$8.91e-08$	$3.55e-07$	$7.17e-08$	$3.49e-07$

TABLE 5.3

Shown are the L^2 and L^∞ errors for a function $u_0(x, y) = f_2(x)f_2(y)$ that has discontinuous derivatives $\frac{\partial u_0}{\partial x}$ and $\frac{\partial u_0}{\partial y}$. The 2D LSIAC-MRA procedure was performed on a series of three meshes. Errors were calculated by excluding the polluted regions around the discontinuous derivatives. We observe that the LSIAC-MRA is effective for error reduction in unpolluted regions.

N	Projection Error		Enhanced Once		Enhanced Each Refinement	
	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error
	\mathbb{P}^0					
70	$1.78e-02$	$5.87e-02$	$1.78e-02$	$5.87e-02$	$1.78e-02$	$5.87e-02$
140	$1.78e-02$	$5.87e-02$	$8.71e-03$	$2.92e-02$	$8.71e-03$	$2.92e-02$
280	$1.78e-02$	$5.87e-02$	$8.71e-03$	$2.92e-02$	$4.44e-03$	$1.47e-02$
	\mathbb{P}^1					
70	$3.63e-04$	$2.27e-03$	$3.63e-04$	$2.27e-03$	$3.63e-04$	$2.27e-03$
140	$3.63e-04$	$2.27e-03$	$9.10e-05$	$5.23e-04$	$9.10e-05$	$5.23e-04$
280	$3.63e-04$	$2.27e-03$	$9.10e-05$	$5.23e-04$	$2.40e-05$	$1.15e-04$
	\mathbb{P}^2					
70	$5.10e-06$	$2.89e-05$	$5.10e-06$	$2.89e-05$	$5.10e-06$	$2.89e-05$
140	$5.10e-06$	$2.89e-05$	$5.59e-07$	$3.30e-06$	$5.59e-07$	$3.30e-06$
280	$5.10e-06$	$2.89e-05$	$5.59e-07$	$3.30e-06$	$1.34e-07$	$8.75e-07$
	\mathbb{P}^3					
70	$5.86e-08$	$4.59e-07$	$5.86e-08$	$4.59e-07$	$5.86e-08$	$4.59e-07$
140	$5.86e-08$	$4.59e-07$	$4.09e-09$	$3.07e-08$	$4.09e-09$	$3.07e-08$
280	$5.86e-08$	$4.59e-07$	$4.09e-09$	$3.07e-08$	$1.77e-09$	$8.27e-09$
	\mathbb{P}^4					
70	$5.12e-10$	$3.12e-09$	$5.12e-10$	$3.12e-09$	$5.12e-10$	$3.12e-09$
140	$5.12e-10$	$3.12e-09$	$4.55e-11$	$2.42e-10$	$4.55e-11$	$2.42e-10$
280	$5.12e-10$	$3.12e-09$	$4.55e-11$	$2.42e-10$	$3.43e-11$	$1.73e-10$

2D L-SIAC Errors in refinement

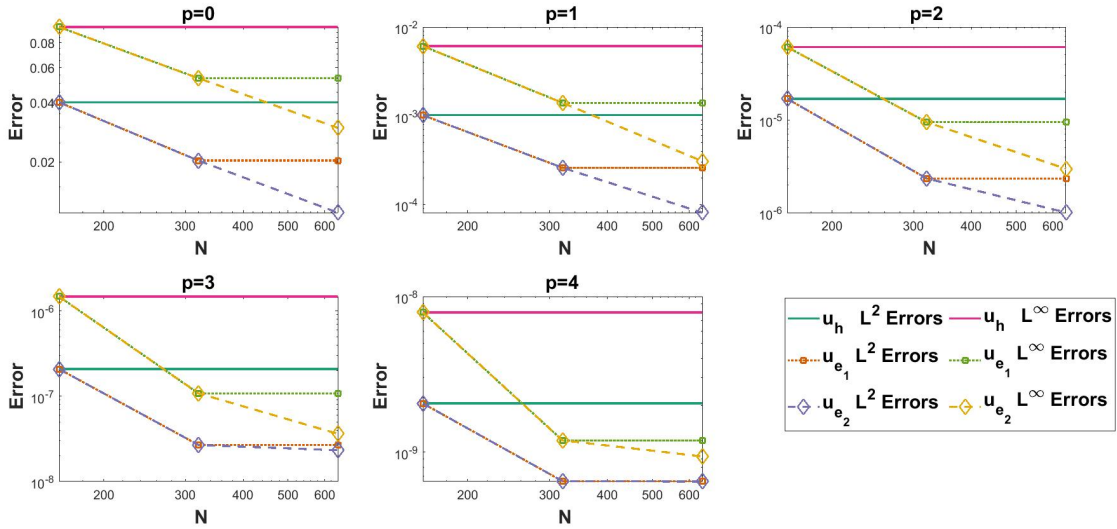


FIG. 5.5. Log-log plots of N vs. Error for a high frequency function $u_0(x, y) = \sin(10\pi x)\sin(10\pi y)$ with wavenumber $k = 5$ on a series of three meshes using 2D LSIAC-MRA. We see that the effectiveness somewhat degrades with increasing polynomial order.

Unpolluted 2D L-SIAC Errors in refinement

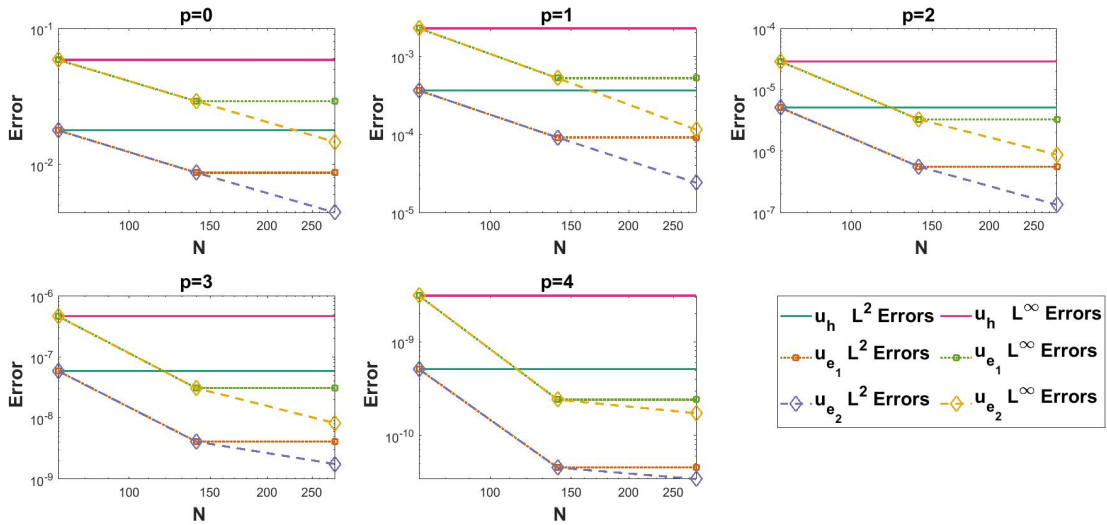


FIG. 5.6. Log-log plots of N vs. Error for a continuous function that has discontinuous $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ derivatives on a series of three meshes using 2D LSIAC-MRA. Errors were calculated by excluding the polluted regions around the discontinuous derivatives. Notice that performance is better than for the discontinuous function.

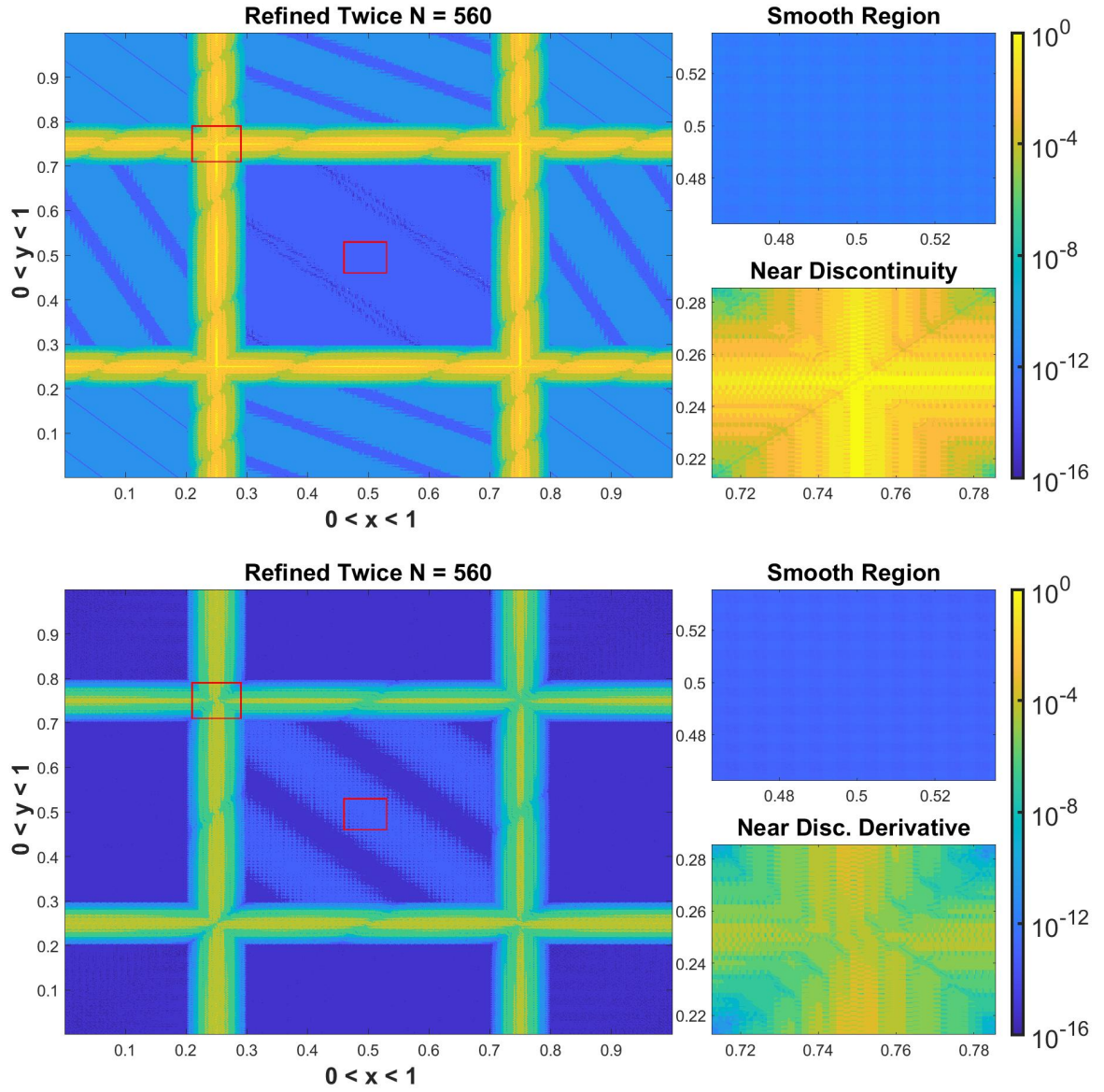


FIG. 5.7. Zoomed in contour error plots for the twice enhanced $p = 4$ approximation of a discontinuous function (Top) and a continuous function that is discontinuous in its $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ derivatives (Bottom). Both approximations were initialized on a 140×140 mesh.

5.2. 3D Test Problems. We now investigate the application of 3D LSIAC-MRA for three-dimensional functions. We have tested the procedure on other functions as well, but present results only for the function $u_0(x, y, z) = \sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z)$ on the domain $\Omega = \{(x, y) \in [0, 1]^3\}$.

Beginning with a projection of this function onto a piecewise orthonormal Legendre basis on a $15 \times 15 \times 15$ uniform hexahedral mesh, the approximation is then enhanced using the LSIAC-MRA procedure. This procedure is applied two times, resulting in a final approximation on a $60 \times 60 \times 60$ mesh. The L^2 - and L^∞ -errors of those approximations with and without the filtering enhancement are given in Table 5.4. Notice that we have included the results for a piecewise constant approximation, $p = 0$. The theory of (L)SIAC does not generally extend to piecewise constants. However, it is evident from the results in these tables that the LSIAC-MRA procedure is effective for piecewise constant approximations. For piecewise linear approximations, $p = 1$, we can see that it is less effective, but still provides an error reduction. Similar to the two-dimensional case, we speculate that for higher degree polynomial approximations it is necessary to start with a higher resolution in order for the LSIAC-MRA procedure to be more effective. The log-log plots for the results of the 3D LSIAC-MRA procedure are given in Figure 5.8.

TABLE 5.4

Table of L^2 and L^∞ errors for initial condition $u(x, y, z) = \sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z)$ on a series of three meshes using 3D LSIAC-MRA. We observe error reduction with each enhancement.

N	Projection Error		Enhanced Once		Enhanced Each Refinement	
	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error	L^2 -Error	L^∞ -Error
	p^0					
15	$1.48e-01$	$6.13e-01$	$1.48e-01$	$6.13e-01$	$1.48e-01$	$6.13e-01$
30	$1.48e-01$	$6.13e-01$	$7.61e-02$	$3.08e-01$	$7.61e-02$	$3.08e-01$
60	$1.48e-01$	$6.13e-01$	$7.61e-02$	$3.08e-01$	$4.43e-02$	$1.59e-01$
	p^1					
15	$7.99e-03$	$4.14e-02$	$7.99e-03$	$4.14e-02$	$7.99e-03$	$4.14e-02$
30	$7.99e-03$	$4.14e-02$	$2.13e-03$	$1.53e-02$	$2.13e-03$	$1.53e-02$
60	$7.99e-03$	$4.14e-02$	$2.13e-03$	$1.53e-02$	$7.70e-04$	$7.53e-03$

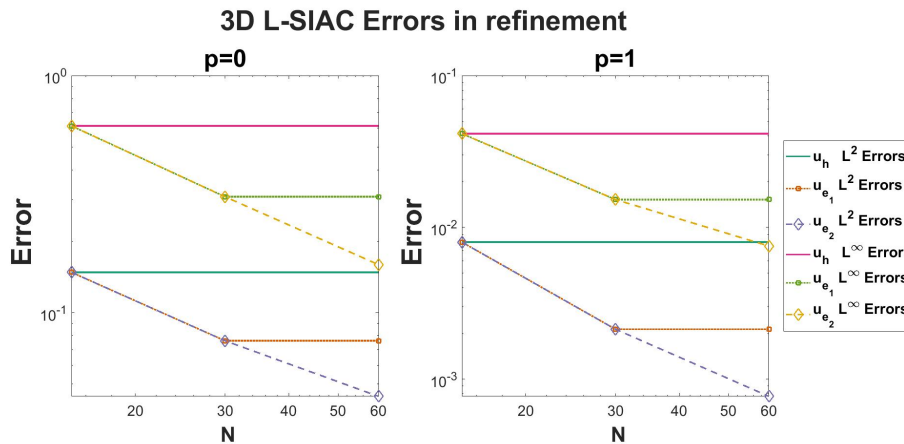


FIG. 5.8. Log-log plots of N vs. Error for the function $u_0(x, y, z) = \sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z)$ on a series of three meshes using 3D LSIAC-MRA. Notice that procedure improves the approximation and provides error reduction.

6. Conclusions and Future Work. In this article, we have introduced an improved multi-resolution analysis scheme for multi-dimensional applications, LSIAC-MRA. This scheme utilizes the Line Smoothness-Increasing Accuracy-Conserving filter which post-processes multi-dimensional data using a one-dimensional support. This allows for approximating the difference coefficients in the multi-wavelet representation and allows for error reduction with mesh refinement. We have provided the underlying operational framework and demonstrated that LSIAC-MRA is effective for two- and three-dimensional applications. Furthermore, LSIAC-MRA is shown to be effective for multi-dimensional piecewise constant data as well. Though the method does not have translational invariance with respect to arbitrary translations owing to the discrete projection, it is invariant with respect to translations by the uniform mesh scaling, h . In the future, we plan on investigating the requirements for the points per wavelength of the initial data as well as allowed non-uniformities in the data in order to leverage LSIAC-MRA for turbulence modeling data or improved identification of undersampled signals.

7. Acknowledgements. This work is partially supported by AFOSR under grant number FA9550-20-1-0166. We would like to thank Dr. Ayaboe Edoh and Dr. Julia Docampo-Sánchez for their valuable comments in the presentation of this paper. Finally, we would like to thank the reviewers for their useful comments that improved the clarity of the presentation of the material contained in this paper.

Statement of Contribution. The ideas in this article were conceived by Jennifer Ryan, who is also responsible for the presentation of information, writing the introduction and conclusion sections, and final edits. Matt Picklo performed the numerical experiments and wrote the majority of the article.

REFERENCES

- [1] A. HARTEN, *Discrete multi-resolution analysis and generalized wavelets*, Appl. Numer. Math., 12 (1993), pp. 153–192, [https://doi.org/https://doi.org/10.1016/0168-9274\(93\)90117-A](https://doi.org/https://doi.org/10.1016/0168-9274(93)90117-A), <https://www.sciencedirect.com/science/article/pii/016892749390117A>. SPECIAL ISSUE.
- [2] B. K. ALPERT, *A class of bases in L^2 for the sparse representation of integral operators*, SIAM J. Math. Anal., 24 (1993), pp. 246–262.
- [3] B. K. ALPERT, G. BEYLKIN, D. GINES, AND L. VOZOVOI, *Adaptive solution of partial differential equations in multiwavelet bases*, J. Comput. Phys., 182 (2002), pp. 149–190.
- [4] A. BOGGESS AND F. J. NARCOWICH, *A First Course in Wavelets with Fourier Analysis*, Prentice Hall, Upper Saddle, New Jersey, 2001.
- [5] C. DE BOOR, *A Practical Guide to Splines*, vol. 27 of Applied mathematical sciences, Springer New York, 1978.
- [6] B. COCKBURN, M. LUSKIN, C.-W. SHU, AND E. SULI, *Enhanced accuracy by post-processing for finite element methods for hyperbolic equations*, Math. Comp., 72 (2003), pp. 577–606.
- [7] A. COHEN, *Numerical Analysis of Wavelet Methods*, Elsevier, Amsterdam, 2003.
- [8] A. DEDNER, J. GIESSELMANN, T. PRYER, AND J. K. RYAN, *Residual estimates for post-processors in elliptic problems*, J. Sci. Comput., 88 (2021).
- [9] J. DOCAMPO-SÁNCHEZ, G. B. JACOBS, X. LI, AND J. K. RYAN, *Enhancing accuracy with a convolution filter: What works and why!*, Computers and Fluids, 213 (2020).
- [10] J. DOCAMPO-SÁNCHEZ, J. K. RYAN, M. MIRZARGAR, AND R. M. KIRBY, *Multi-dimensional filtering: Reducing the dimension through rotation*, SIAM J. Sci. Comput., 39 (2017), pp. A2179–A2200.
- [11] N. HOVHANNISYAN, S. MÜLLER, AND R. SCHÄFER, *Adaptive multiresolution discontinuous galerkin schemes for conservation laws*, Report 311, Institut für Geometrie und Praktische Mathematik, Aachen, sep 2010. <http://www.igpm.rwth-aachen.de/en/reports2010>.
- [12] L. JI, P. VAN SLINGERLAND, J. K. RYAN, AND K. VUIK, *Superconvergent error estimates for a position-dependent smoothness-increasing accuracy-conserving filter for DG solutions*, Math. Comp., 83 (2014), pp. 2239–2262.
- [13] L. JI, Y. XU, AND J. K. RYAN, *Negative-order norm estimates for nonlinear hyperbolic conservation laws*, J. Sci. Comput., 54 (2013), pp. 269–310.
- [14] K. SCHNEIDER AND O. VASILYEV, *Wavelet methods in computational fluid dynamics*, Annual Review of Fluid Mechanics, 42 (2010), pp. 473–503, <https://doi.org/10.1146/annurev-fluid-121108-145637>.
- [15] X. MENG AND J. K. RYAN, *Discontinuous galerkin methods for nonlinear scalar hyperbolic conservation laws: Divided difference estimates and accuracy enhancement*, Numer. Math., 136 (2017), pp. 27–73.
- [16] E. J. NYSTRÖM, *über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben*, Acta Numer., 54 (1930), pp. 185–204, <https://doi.org/doi:10.1007/BF02547521>.
- [17] J. K. RYAN, *Capitalizing on superconvergence for more accurate multi-resolution discontinuous galerkin methods*, Communications in Applied Mathematics and Computation, (2021).
- [18] J. K. RYAN AND B. COCKBURN, *Local derivative post-processing for the discontinuous galerkin method*, J. Comput. Phys., 228 (2009), pp. 8642–8664.
- [19] J. K. RYAN, X. LI, R. M. KIRBY, AND K. VUIK, *One-sided position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering over uniform and non-uniform meshes*, J. Sci. Comput., 64 (2015), pp. 773–817.

- [20] J. K. RYAN, C.-W. SHU, AND H. ATKINS, *Extension of a post-processing technique for the discontinuous galerkin method for hyperbolic equations with application to an aeroacoustic problem*, SIAM J. Sci. Comput., 26 (2005), pp. 821–843.
- [21] S. G. MALLAT, *A theory for multiresolution signal decomposition: The wavelet representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 674–693, <https://doi.org/10.1109/34.192463>.
- [22] S. MÜLLER, *Adaptive Multiscale Schemes for Conservation Laws*, Springer Berlin Heidelberg, 2003, <https://doi.org/10.1007/978-3-642-18164-1>, <https://doi.org/10.1007%2F978-3-642-18164-1>.
- [23] P. VAN SLINGERLAND, J. K. RYAN, AND K. VUIK, *Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for accuracy for improving discontinuous galerkin solutions*, SIAM J. Sci. Comput., 33 (2011), pp. 802–825.
- [24] W. DAHMEN, A. KUNOTH, AND K. URBAN, *A wavelet galerkin method for the stokes equations*, Computing, 56 (1996), pp. 259–301, <https://doi.org/10.1007/BF02238515>.
- [25] Y. MEYER, *Ondelettes et fonctions splines*, Séminaire Équations aux dérivées partielles (Polytechnique) dit aussi "Séminaire Goulaouic-Schwartz", (1986-1987), http://www.numdam.org/item/SEDP_1986-1987____A5_0/. talk:6.

Supplementary Materials

March 11, 2022

S1 SIAC-MRA: Filtered Basis Functions & Coefficients:

In these supplementary materials we detail the χ -functions and associated coefficients appearing in the simplified filtered approximations. In addition, figures are provided delineating the different domains of definition for these functions. Denote by ζ the mapping of the filtering point to the reference element $[-1, 1]$, i.e. $\zeta_x = \frac{2}{h}(x - x_i)$, $\zeta_y = \frac{2}{h}(y - y_j)$, $\zeta_z = \frac{2}{h}(z - z_k)$.

S1.1 One-dimensional SIAC-MRA:

We first recall that the filtered approximation given in Ryan, (2021) is of the form

$$u_h^*(x) = \begin{cases} \sum_{m=0}^{p+1} d_m^j \chi_L^m(\zeta_x), & \zeta_x \in (-1, 0) \\ \sum_{m=0}^{p+1} \tilde{d}_m^j \chi_R^m(\zeta_x), & \zeta_x \in (0, 1) \end{cases},$$

with the χ functions and coefficients given in Table S1.

Table S1: Table of basis functions and associated coefficients resulting from the analytical evaluation of the convolution in $u_h^*(x)$.

\mathbf{k}	$\mathbf{d}_{\mathbf{k}}^j$	$\tilde{\mathbf{d}}_{\mathbf{k}}^j$
$\mathbf{0}$	$\frac{\sqrt{2}}{2} \{ \sum_{\gamma=-p}^p c_\gamma u_0^{j+\gamma} \}$	$\frac{\sqrt{2}}{2} \{ \sum_{\gamma=-p}^p c_\gamma u_0^{j+\gamma} \}$
$\mathbf{1}$	$\frac{\sqrt{2}}{4} \{ \sum_{\gamma=-p}^p c_\gamma (u_0^{j+\gamma} - u_0^{j+\gamma-1}) \}$	$\frac{\sqrt{2}}{4} \{ \sum_{\gamma=-p}^p c_\gamma (u_0^{j+\gamma+1} - u_0^{j+\gamma}) \}$
$\mathbf{k} > \mathbf{1}$	$\frac{1}{2} \{ \sum_{\gamma=-p}^p c_\gamma (u_k^{j+\gamma} - u_k^{j+\gamma-1}) \}$	$\frac{1}{2} \{ \sum_{\gamma=-p}^p c_\gamma (u_k^{j+\gamma+1} - u_k^{j+\gamma}) \}$
	$\chi_{\mathbf{L}}^{\mathbf{k}}(\zeta)$	$\chi_{\mathbf{R}}^{\mathbf{k}}(\zeta)$
$\mathbf{0.}$	1	1
$\mathbf{1.}$	ζ	ζ
$\mathbf{k} > \mathbf{1.}$	$\frac{\sqrt{k-1/2}}{2k-1} [P_k(1 + \zeta) - P_{k-2}(1 + \zeta)]$	$\frac{\sqrt{k-1/2}}{2k-1} [P_k(\zeta - 1) - P_{k-2}(\zeta - 1)]$

S1.2 Two-dimensional LSIAC-MRA

There are seven different forms for the coefficients included in the two-dimensional L-SIAC case with the basis given in Table S2. These coefficients are of the form

$$a_{m,r}^{S,P} = a_{m,r}(q_x, q_y) = \sum_{\gamma=-p}^p c_\gamma u_{m,r}^{i+\gamma+qx, j+\gamma+qy},$$

where the arguments (q_x, q_y) are determined by the region S and position P of the filter point. A table of these dependencies by region and position is provided in Table S3. The regions themselves are depicted in Figure S1. An algorithm for the LSIAC-MRA procedure is included in Algorithm S1.

Table S2: 2D L-SIAC basis functions. For notational convenience suppose \mathbf{x} belongs to element (i, j) .

Region S	Basis Function	Definition
1T	$\chi_{1T,L}^{m,r}(\zeta_x, \zeta_y)$	$\frac{2-\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\zeta_x - \frac{\zeta_y}{2} + \left(1 - \frac{\zeta_y}{2}\right)\eta \right) \phi_{ky} \left(\left(1 - \frac{\zeta_y}{2}\right)\eta + \frac{\zeta_y}{2} \right) d\eta$
	$\chi_{1T,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_y - \zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_y - \zeta_x}{2}\eta + \frac{\zeta_x - \zeta_y}{2} + 1 \right) \phi_{ky} \left(\frac{\zeta_y - \zeta_x}{2}\eta + \frac{\zeta_y - \zeta_x}{2} - 1 \right) d\eta$
	$\chi_{1T,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_x}{2}\eta + \frac{\zeta_x}{2} - 1 \right) \phi_{ky} \left(\frac{\zeta_x}{2}\eta + \zeta_y - 1 - \frac{\zeta_x}{2} \right) d\eta$
1B	$\chi_{1B,L}^{m,r}(\zeta_x, \zeta_y)$	$\frac{2-\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\left(1 - \frac{\zeta_x}{2}\right)\eta + \frac{\zeta_x}{2} \right) \phi_{ky} \left(\zeta_y - \frac{\zeta_x}{2} + \left(1 - \frac{\zeta_x}{2}\right)\eta \right) d\eta$
	$\chi_{1B,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_x - \zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_x - \zeta_y}{2}\eta + \frac{\zeta_x - \zeta_y}{2} - 1 \right) \phi_{ky} \left(\frac{\zeta_x - \zeta_y}{2}\eta + \frac{\zeta_y - \zeta_x}{2} + 1 \right) d\eta$
	$\chi_{1B,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_y}{2}\eta + \zeta_x - \frac{\zeta_y}{2} - 1 \right) \phi_{ky} \left(\frac{\zeta_y}{2}\eta - 1 + \frac{\zeta_y}{2} \right) d\eta$
2T	$\chi_{2T,L}^{m,r}(\zeta_x, \zeta_y)$	$-\frac{\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(-\frac{\zeta_x}{2}\eta + \frac{\zeta_x}{2} + 1 \right) \phi_{ky} \left(-\frac{\zeta_x}{2}\eta + \zeta_y - 1 - \frac{\zeta_x}{2} \right) d\eta$
	$\chi_{2T,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{2+\zeta_x-\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\frac{2+\zeta_x-\zeta_y}{2}\eta + \frac{\zeta_x-\zeta_y}{2} \right) \phi_{ky} \left(\frac{2+\zeta_x-\zeta_y}{2}\eta + \frac{\zeta_y-\zeta_x}{2} \right) d\eta$
	$\chi_{2T,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_y}{2}\eta + \zeta_x - \frac{\zeta_y}{2} + 1 \right) \phi_{ky} \left(\frac{\zeta_y}{2}\eta + \frac{\zeta_y}{2} - 1 \right) d\eta$
3T	$\chi_{3T,L}^{m,r}(\zeta_x, \zeta_y)$	$-\frac{\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(-\frac{\zeta_y}{2}\eta - \frac{\zeta_y}{2} + \zeta_x + 1 \right) \phi_{ky} \left(-\frac{\zeta_y}{2}\eta + 1 + \frac{\zeta_y}{2} \right) d\eta$
	$\chi_{3T,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_y - \zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_y - \zeta_x}{2}\eta + \frac{\zeta_x - \zeta_y}{2} + 1 \right) \phi_{ky} \left(\frac{\zeta_y - \zeta_x}{2}\eta + \frac{\zeta_y - \zeta_x}{2} - 1 \right) d\eta$
	$\chi_{3T,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{2+\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\left(\frac{2+\zeta_x}{2}\right)\eta + \frac{\zeta_x}{2} \right) \phi_{ky} \left(\left(\frac{2+\zeta_x}{2}\right)\eta - \frac{\zeta_x}{2} + \zeta_y \right) d\eta$
3B	$\chi_{3B,L}^{m,r}(\zeta_x, \zeta_y)$	$-\frac{\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(-\frac{\zeta_x}{2}\eta + \frac{\zeta_x}{2} + 1 \right) \phi_{ky} \left(-\frac{\zeta_x}{2}\eta + 1 + \zeta_y - \frac{\zeta_x}{2} \right) d\eta$
	$\chi_{3B,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_x - \zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_x - \zeta_y}{2}\eta + \frac{\zeta_x - \zeta_y}{2} - 1 \right) \phi_{ky} \left(\frac{\zeta_x - \zeta_y}{2}\eta + \frac{\zeta_y - \zeta_x}{2} + 1 \right) d\eta$
	$\chi_{3B,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{2+\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(\left(\frac{2+\zeta_y}{2}\right)\eta + \zeta_x - \frac{\zeta_y}{2} \right) \phi_{ky} \left(\left(\frac{2+\zeta_y}{2}\right)\eta + \frac{\zeta_y}{2} \right) d\eta$
4B	$\chi_{4B,L}^{m,r}(\zeta_x, \zeta_y)$	$-\frac{\zeta_y}{2} \int_{-1}^1 \phi_{kx} \left(-\frac{\zeta_y}{2}\eta - \frac{\zeta_y}{2} + \zeta_x - 1 \right) \phi_{ky} \left(-\frac{\zeta_y}{2}\eta + 1 + \frac{\zeta_y}{2} \right) d\eta$
	$\chi_{4B,M}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_y - \zeta_x + 2}{2} \int_{-1}^1 \phi_{kx} \left(\left(1 + \frac{\zeta_y - \zeta_x}{2}\right)\eta + \frac{\zeta_x - \zeta_y}{2} \right) \phi_{ky} \left(\left(1 + \frac{\zeta_y - \zeta_x}{2}\right)\eta + \frac{\zeta_y - \zeta_x}{2} \right) d\eta$
	$\chi_{4B,R}^{m,r}(\zeta_x, \zeta_y)$	$\frac{\zeta_x}{2} \int_{-1}^1 \phi_{kx} \left(\frac{\zeta_x}{2}\eta + \frac{\zeta_x}{2} - 1 \right) \phi_{ky} \left(\frac{\zeta_x}{2}\eta + 1 + \zeta_y - \frac{\zeta_x}{2} \right) d\eta$

Table S3: Table describing how the shifting of the arguments of the a coefficients vary by region.

Region S	Arguments of $a_\alpha(q_x, q_y)$	$\mathbf{P} = \mathbf{L}$	$\mathbf{P} = \mathbf{M}$	$\mathbf{P} = \mathbf{R}$
1T		(0, 0)	(0, 1)	(1, 1)
1B		(0, 0)	(1, 0)	(1, 1)
2T		(-1, 0)	(0, 0)	(0, 1)
3T		(-1, -1)	(-1, 0)	(0, 0)
3B		(-1, -1)	(0, -1)	(0, 0)
4B		(0, -1)	(0, 0)	(1, 0)

33

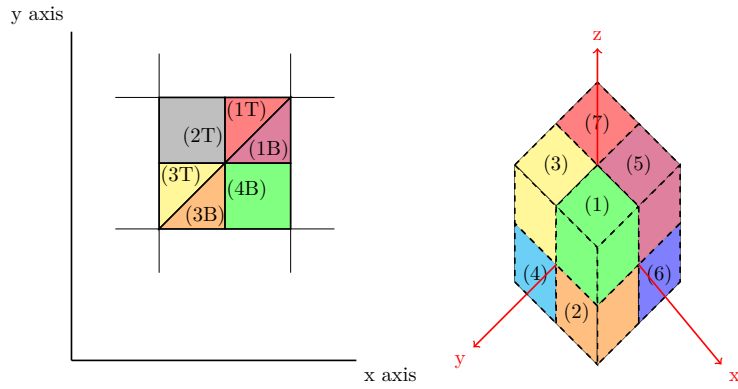


Figure S1: Partition of domain for both 2D L-SIAC (left) and the octant numbering scheme in 3D (right).

S1.3 Three-dimensional LSIAC-MRA

In three-dimensions, the tensor product expansion functions are of the form

$$\chi_{S,P}^{m,r,\ell} = C \int_{-1}^1 \phi_m(\arg_x) \phi_r(\arg_y) \phi_\ell(\arg_z)$$

where the row of the table determines the Region S and Position $P = \{L, LM, RM, R\}$. The regions are described in Tables S4 and S5 with visual depictions in Figures S1 and S2. The coefficient C is given in the coefficient column, and the arguments of the Legendre polynomials are given in the appropriate argument column. These can be found in Table S6. There are fifteen different forms for the coefficients included in the 3D L-SIAC case. These coefficients are of the form

$$a_{m,r,\ell}^{S,P} = a_{m,r,\ell}(q_x, q_y, q_z) = \sum_{\gamma=-p}^p c_\gamma u_{m,r,\ell}^{i+\gamma+qx, j+\gamma+qy, k+\gamma+qz},$$

where the coefficients (q_x, q_y, q_z) are determined by the region S and position P of the filter point. A table of these dependencies by region and position is provided in Table S7.

Table S4: Regions present by octant

Octant	Regions Present
1	1A, 1B, 1C, 1D, 1E, 1F
2	2E, 2F
3	3A, 3B
4	4B, 4F
5	5C, 5D
6	6D, 6E
7	7A, 7C
8	8A, 8B, 8C, 8D, 8E, 8F

Table S5: Classification of regions.

Class	Filtering Coordinate Inequality
(A)	$\zeta_x \leq \zeta_y \leq \zeta_z$
(B)	$\zeta_x \leq \zeta_z < \zeta_y$
(C)	$\zeta_y < \zeta_x \leq \zeta_z$
(D)	$\zeta_y \leq \zeta_z < \zeta_x$
(E)	$\zeta_z < \zeta_y \leq \zeta_x$
(F)	$\zeta_z < \zeta_x < \zeta_y$

Algorithm S1 Calculate u^f from u using L-SIAC kernel

```

1: Input: Number of elements in one direction  $N$ , and polynomial degree  $p$ .
2: Output: Fine mesh modal coefficients  $u^f$ 
3:
4: % Compute modal projection on coarse mesh
5:  $u = \text{Initialize}(N, p, \Omega, u_0)$ 
6:
7: % Compute projection matrices by region
8:  $[P1TL, P1TM, P1TR] = \text{ProjMatrixL}(N, p, Q1T)$ 
9:  $[P1BL, P1BM, P1BR] = \text{ProjMatrixL}(N, p, Q1B)$ 
10:  $[P2TL, P2TM, P2TR] = \text{ProjMatrixL}(N, p, Q2T)$ 
11:  $[P3TL, P3TM, P3TR] = \text{ProjMatrixL}(N, p, Q3T)$ 
12:  $[P3BL, P3BM, P3BR] = \text{ProjMatrixL}(N, p, Q3B)$ 
13:  $[P4BL, P4BM, P4BR] = \text{ProjMatrixL}(N, p, Q4B)$ 
14:
15: % Loop through mesh elements
16: for  $i \leftarrow 1, N$  do
17:   for  $j \leftarrow 1, N$  do
18:
19:     % Construct filtering/sifting matrices
20:      $K11 = \text{FiltMatrixL}(p, 1, 1, i, j)$ 
21:      $K01 = \text{FiltMatrixL}(p, 0, 1, i, j)$ 
22:      $K10 = \text{FiltMatrixL}(p, 1, 0, i, j)$ 
23:      $K00 = \text{FiltMatrixL}(p, 0, 0, i, j)$ 
24:      $Kn10 = \text{FiltMatrixL}(p, -1, 0, i, j)$ 
25:      $K0n1 = \text{FiltMatrixL}(p, 0, -1, i, j)$ 
26:      $Kn1n1 = \text{FiltMatrixL}(p, -1, -1, i, j)$ 
27:
28:     % Compute local fine mesh coefficients
29:      $u_{tensor}^f(:, 2i, 2j) =$ 
         $((P1TL + P1BL) * K00 +$ 
         $(P1TR + P1BR) * K11 +$ 
         $(P1TM * K01 + P1BM * K10) * u$ 
30:      $u_{tensor}^f(:, 2i - 1, 2j) =$ 
         $(P2L * Kn10 + P2M * K00 + P2R * K01) * u$ 
31:      $u_{tensor}^f(:, 2i - 1, 2j - 1) =$ 
         $((P3TL + P3BL) * Kn1n1 +$ 
         $(P3TR + P3BR) * K00 +$ 
         $P3TM * Kn10 + P3BM * K0n1) * u$ 
32:      $u_{tensor}^f(:, 2i, 2j - 1) =$ 
         $(P4L * K0n1 + P4M * K00 + P4R * K10) * u$ 
33:   end for
34: end for
35:
36: % Reshape fine modal array into original array form
37:  $u^f = \text{ReshapeArray}(p, N, u_{tensor}^f)$ 

```

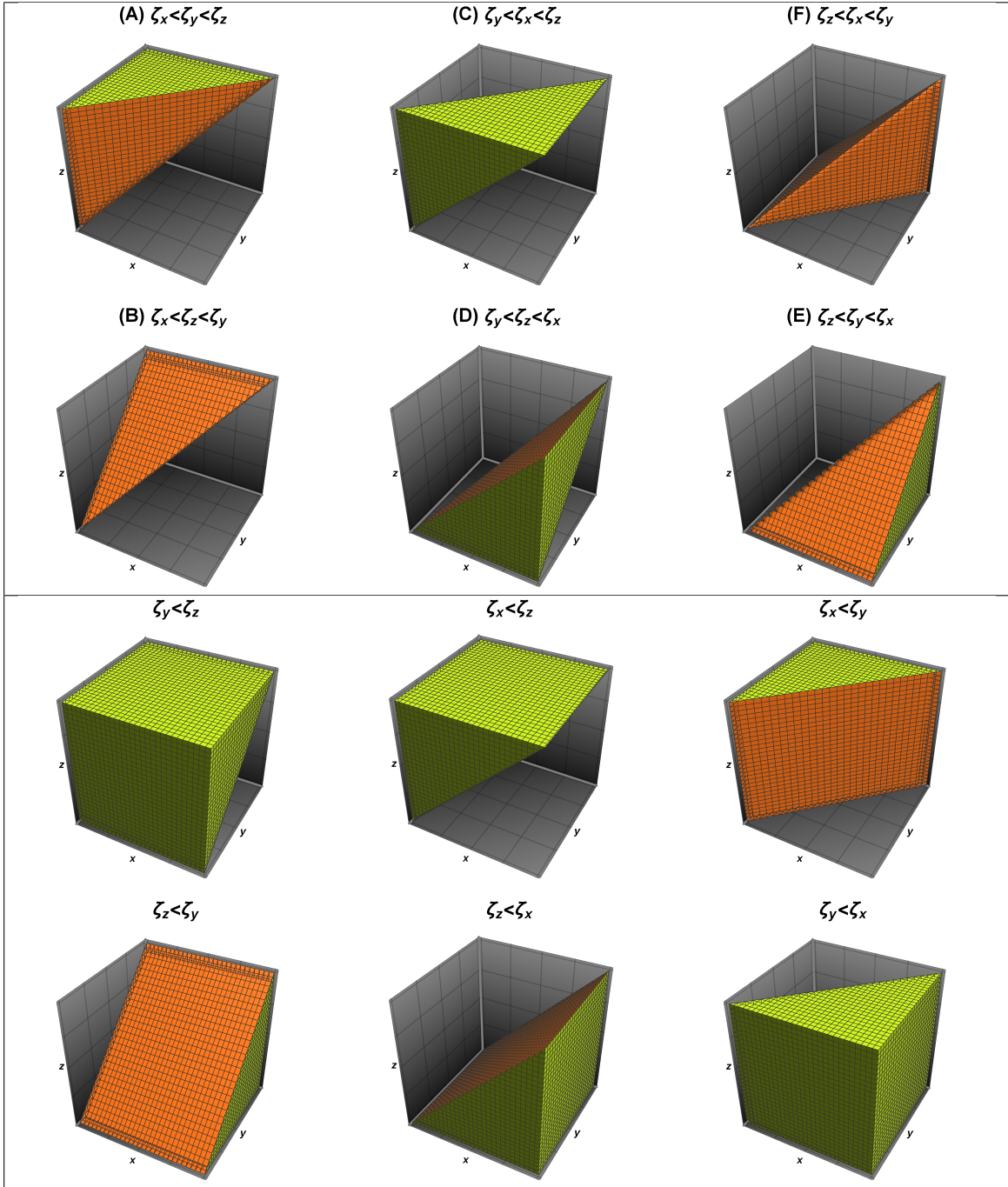


Figure S2: Depiction of how the domain of dependence of the filtered approximation form varies from octant to octant. Octants 1 and 8 have six regions (Top), while octants 2 through 7 have only a pair of regions each (Bottom). The form of these regions is a consequence of the filtering coordinate inequalities given in Table S5 coupled with the relative location of each octant in the reference element.

Table S6: Three-dimensional L-SIAC basis functions. For notational convenience suppose that \mathbf{x} belong to element (i, j, k) and that the region S and position P of the function are as indicated in the table rows.

Region S	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1A	$\chi^{S,L}$	$(2 - \zeta_z)/4$	$w + \zeta_x - \zeta_z/2 - (w * \zeta_z)/2$	$w + \zeta_y - \zeta_z/2 - (w * \zeta_z)/2$	$w - ((-1 + w) * \zeta_z)/2$
1A	$\chi^{S,LM}$	$(-\zeta_y + \zeta_z)/4$	$(2 + 2 * \zeta_x - (1 + w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1 + w) * \zeta_z)/2$	$(-2 - (1 + w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
1A	$\chi^{S,RM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1 + w) * \zeta_y)/2$	$(-2 - (1 + w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$-1 - ((1 + w) * \zeta_x)/2 + ((-1 + w) * \zeta_y)/2 + \zeta_z$
1A	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$-1 + ((-1 + w) * \zeta_x)/2 + \zeta_y$	$-1 + ((-1 + w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1B	$\chi^{S,L}$	$(2 - \zeta_y)/4$	$w + \zeta_x - \zeta_y/2 - (w * \zeta_y)/2$	$(-w * (-2 + \zeta_y) + \zeta_y)/2$	$w - \zeta_y/2 - (w * \zeta_y)/2 + \zeta_z$
1B	$\chi^{S,LM}$	$(\zeta_y - \zeta_z)/4$	$(2 + 2 * \zeta_x + (-1 + w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1 + w) * \zeta_y - (1 + w) * \zeta_z)/2$	$(2 + (-1 + w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
1B	$\chi^{S,RM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1 + w) * \zeta_z)/2$	$(-2 - (1 + w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1 + w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
1B	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$-1 + ((-1 + w) * \zeta_x)/2 + \zeta_y$	$-1 + ((-1 + w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1C	$\chi^{S,L}$	$(2 - \zeta_z)/4$	$w + \zeta_x - \zeta_z/2 - (w * \zeta_z)/2$	$w + \zeta_y - \zeta_z/2 - (w * \zeta_z)/2$	$w - ((-1 + w) * \zeta_z)/2$
1C	$\chi^{S,LM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1 + w) * \zeta_z)/2$	$(2 - (1 + w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1 + w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
1C	$\chi^{S,RM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1 + w) * \zeta_x - (1 + w) * \zeta_y)/2$	$(2 + (-1 + w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$-1 + ((-1 + w) * \zeta_x)/2 - ((1 + w) * \zeta_y)/2 + \zeta_z$
1C	$\chi^{S,R}$	$\zeta_y/4$	$-1 + \zeta_x + ((-1 + w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$-1 + ((-1 + w) * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1D	$\chi^{S,L}$	$(2 - \zeta_x)/4$	$(-w * (-2 + \zeta_x) + \zeta_x)/2$	$w - \zeta_x/2 - (w * \zeta_x)/2 + \zeta_y$	$w - \zeta_x/2 - (w * \zeta_x)/2 + \zeta_z$
1D	$\chi^{S,LM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1 + w) * \zeta_x - (1 + w) * \zeta_z)/2$	$(2 + (-1 + w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1 + w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
1D	$\chi^{S,RM}$	$(-\zeta_y + \zeta_z)/4$	$(-2 + 2 * \zeta_x - (1 + w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1 + w) * \zeta_z)/2$	$(-2 - (1 + w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
1D	$\chi^{S,R}$	$\zeta_y/4$	$-1 + \zeta_x + ((-1 + w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$-1 + ((-1 + w) * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1E	$\chi^{S,L}$	$(2 - \zeta_x)/4$	$(-w * (-2 + \zeta_x) + \zeta_x)/2$	$w - \zeta_x/2 - (w * \zeta_x)/2 + \zeta_y$	$w - \zeta_x/2 - (w * \zeta_x)/2 + \zeta_z$
1E	$\chi^{S,LM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1 + w) * \zeta_x - (1 + w) * \zeta_y)/2$	$(2 + (-1 + w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$1 + ((-1 + w) * \zeta_x)/2 - ((1 + w) * \zeta_y)/2 + \zeta_z$
1E	$\chi^{S,RM}$	$(\zeta_y - \zeta_z)/4$	$(-2 + 2 * \zeta_x + (-1 + w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1 + w) * \zeta_y - (1 + w) * \zeta_z)/2$	$(2 + (-1 + w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
1E	$\chi^{S,R}$	$\zeta_z/4$	$-1 + \zeta_x + ((-1 + w) * \zeta_z)/2$	$-1 + \zeta_y + ((-1 + w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$
Region	Basis Function Parameters				
Basis Function	Coefficient	\mathbf{arg}_x	\mathbf{arg}_y	\mathbf{arg}_z	
1F	$\chi^{S,L}$	$(2 - \zeta_y)/4$	$w + \zeta_x - \zeta_y/2 - (w * \zeta_y)/2$	$(-w * (-2 + \zeta_y) + \zeta_y)/2$	$w - \zeta_y/2 - (w * \zeta_y)/2 + \zeta_z$
1F	$\chi^{S,LM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1 + w) * \zeta_y)/2$	$(-2 - (1 + w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$1 - ((1 + w) * \zeta_x)/2 + ((-1 + w) * \zeta_y)/2 + \zeta_z$
1F	$\chi^{S,RM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1 + w) * \zeta_x - (1 + w) * \zeta_z)/2$	$(-2 + (-1 + w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1 + w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
1F	$\chi^{S,R}$	$\zeta_z/4$	$-1 + \zeta_x + ((-1 + w) * \zeta_z)/2$	$-1 + \zeta_y + ((-1 + w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$

Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
2E	$\chi^{S,L}$	$-\zeta_z/4$	$-1 + \zeta_x - ((1+w) * \zeta_z)/2$	$-1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
2E	$\chi^{S,LM}$	$(2 - \zeta_x + \zeta_z)/4$	$(\zeta_x - \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$	$(-\zeta_x + 2 * \zeta_y - \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$	$(-\zeta_x + \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$
2E	$\chi^{S,RM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_y)/2$	$(2 + (-1+w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$1 + ((-1+w) * \zeta_x)/2 - ((1+w) * \zeta_y)/2 + \zeta_z$
2E	$\chi^{S,R}$	$\zeta_y/4$	$-1 + \zeta_x + ((-1+w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$1 + ((-1+w) * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
2F	$\chi^{S,L}$	$-\zeta_z/4$	$-1 + \zeta_x - ((1+w) * \zeta_z)/2$	$-1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
2F	$\chi^{S,LM}$	$(2 - \zeta_y + \zeta_z)/4$	$(2 * \zeta_x - \zeta_y - \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$	$(\zeta_y - \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$	$(-\zeta_y + \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$
2F	$\chi^{S,RM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_y)/2$	$(-2 - (1+w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$1 - ((1+w) * \zeta_x)/2 + ((-1+w) * \zeta_y)/2 + \zeta_z$
2F	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$-1 + ((-1+w) * \zeta_x)/2 + \zeta_y$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
3A	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_z$
3A	$\chi^{S,LM}$	$(2 + \zeta_x - \zeta_z)/4$	$(\zeta_x + w * (2 + \zeta_x - \zeta_z) - \zeta_z)/2$	$(-\zeta_x + 2 * \zeta_y + w * (2 + \zeta_x - \zeta_z) - \zeta_z)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_z) + \zeta_z)/2$
3A	$\chi^{S,RM}$	$(-\zeta_y + \zeta_z)/4$	$(2 + 2 * \zeta_x - (1+w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
3A	$\chi^{S,R}$	$\zeta_y/4$	$1 + \zeta_x + ((-1+w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$-1 + ((-1+w) * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
3B	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_z$
3B	$\chi^{S,LM}$	$(2 + \zeta_x - \zeta_y)/4$	$(\zeta_x + w * (2 + \zeta_x - \zeta_y) - \zeta_y)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_y) + \zeta_y)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_y) - \zeta_y + 2 * \zeta_z)/2$
3B	$\chi^{S,RM}$	$(\zeta_y - \zeta_z)/4$	$(2 + 2 * \zeta_x + (-1+w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1+w) * \zeta_y - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
3B	$\chi^{S,R}$	$\zeta_z/4$	$1 + \zeta_x + ((-1+w) * \zeta_z)/2$	$-1 + \zeta_y + ((-1+w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
4B	$\chi^{S,L}$	$-\zeta_z/4$	$1 + \zeta_x - ((1+w) * \zeta_z)/2$	$-1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
4B	$\chi^{S,LM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
4B	$\chi^{S,RM}$	$(2 + \zeta_x - \zeta_y)/4$	$(\zeta_x + w * (2 + \zeta_x - \zeta_y) - \zeta_y)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_y) + \zeta_y)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_y) - \zeta_y + 2 * \zeta_z)/2$
4B	$\chi^{S,R}$	$\zeta_y/4$	$1 + \zeta_x + ((-1+w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$1 + ((-1+w) * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
4F	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$1 - ((1+w) * \zeta_x)/2 + \zeta_z$
4F	$\chi^{S,LM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_z)/2$	$(-2 + (-1+w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
4F	$\chi^{S,RM}$	$(2 - \zeta_y + \zeta_z)/4$	$(2 * \zeta_x - \zeta_y - \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$	$(\zeta_y - \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$	$(-\zeta_y + \zeta_z + w * (2 - \zeta_y + \zeta_z))/2$
4F	$\chi^{S,R}$	$\zeta_y/4$	$1 + \zeta_x + ((-1+w) * \zeta_y)/2$	$(-2 + \zeta_y + w * \zeta_y)/2$	$1 + ((-1+w) * \zeta_y)/2 + \zeta_z$

Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
5C	$\chi^{S,L}$	$-\zeta_y/4$	$-1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$-1 - ((1+w) * \zeta_y)/2 + \zeta_z$
5C	$\chi^{S,LM}$	$(2 + \zeta_y - \zeta_z)/4$	$(2 * \zeta_x - \zeta_y + w * (2 + \zeta_y - \zeta_z) - \zeta_z)/2$	$(\zeta_y + w * (2 + \zeta_y - \zeta_z) - \zeta_z)/2$	$(-\zeta_y + w * (2 + \zeta_y - \zeta_z) + \zeta_z)/2$
5C	$\chi^{S,RM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_z)/2$	$(2 - (1+w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
5C	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_y$	$-1 + ((-1+w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
5D	$\chi^{S,L}$	$-\zeta_y/4$	$-1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$-1 - ((1+w) * \zeta_y)/2 + \zeta_z$
5D	$\chi^{S,LM}$	$(2 - \zeta_x + \zeta_y)/4$	$(\zeta_x - \zeta_y + w * (2 - \zeta_x + \zeta_y))/2$	$(-\zeta_x + \zeta_y + w * (2 - \zeta_x + \zeta_y))/2$	$(-\zeta_x - \zeta_y + w * (2 - \zeta_x + \zeta_y) + 2 * \zeta_z)/2$
5D	$\chi^{S,RM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
5D	$\chi^{S,R}$	$\zeta_z/4$	$-1 + \zeta_x + ((-1+w) * \zeta_z)/2$	$1 + \zeta_y + ((-1+w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
6D	$\chi^{S,L}$	$-\zeta_z/4$	$-1 + \zeta_x - ((1+w) * \zeta_z)/2$	$1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
6D	$\chi^{S,LM}$	$(-\zeta_y + \zeta_z)/4$	$(-2 + 2 * \zeta_x - (1+w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
6D	$\chi^{S,HM}$	$(2 - \zeta_x + \zeta_y)/4$	$(\zeta_x - \zeta_y + w * (2 - \zeta_x + \zeta_y))/2$	$(-\zeta_x + \zeta_y + w * (2 - \zeta_x + \zeta_y))/2$	$(-\zeta_x - \zeta_y + w * (2 - \zeta_x + \zeta_y) + 2 * \zeta_z)/2$
6D	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_y$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
6E	$\chi^{S,L}$	$-\zeta_y/4$	$-1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$1 - ((1+w) * \zeta_y)/2 + \zeta_z$
6E	$\chi^{S,LM}$	$(\zeta_y - \zeta_z)/4$	$(-2 + 2 * \zeta_x + (-1+w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1+w) * \zeta_y - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
6E	$\chi^{S,RM}$	$(2 - \zeta_x + \zeta_z)/4$	$(\zeta_x - \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$	$(-\zeta_x + 2 * \zeta_y - \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$	$(-\zeta_x + \zeta_z + w * (2 - \zeta_x + \zeta_z))/2$
6E	$\chi^{S,R}$	$\zeta_x/4$	$(-2 + \zeta_x + w * \zeta_x)/2$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_y$	$1 + ((-1+w) * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
7A	$\chi^{S,L}$	$-\zeta_y/4$	$1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$-1 - ((1+w) * \zeta_y)/2 + \zeta_z$
7A	$\chi^{S,LM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_y)/2$	$(-2 - (1+w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$-1 - ((1+w) * \zeta_x)/2 + ((-1+w) * \zeta_y)/2 + \zeta_z$
7A	$\chi^{S,RM}$	$(2 + \zeta_y - \zeta_z)/4$	$(\zeta_x + w * (2 + \zeta_x - \zeta_z) - \zeta_z)/2$	$(-\zeta_x + 2 * \zeta_y + w * (2 + \zeta_x - \zeta_z) - \zeta_z)/2$	$(-\zeta_x + w * (2 + \zeta_x - \zeta_z) + \zeta_z)/2$
7A	$\chi^{S,R}$	$\zeta_z/4$	$1 + \zeta_x + ((-1+w) * \zeta_z)/2$	$1 + \zeta_y + ((-1+w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$
Region	Basis Function Parameters				
	Basis Function	Coefficient	\arg_x	\arg_y	\arg_z
7C	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$-1 - ((1+w) * \zeta_x)/2 + \zeta_z$
7C	$\chi^{S,LM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_y)/2$	$(2 + (-1+w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$-1 + ((-1+w) * \zeta_x)/2 - ((1+w) * \zeta_y)/2 + \zeta_z$
7C	$\chi^{S,RM}$	$(2 + \zeta_y - \zeta_z)/4$	$(2 * \zeta_x - \zeta_y + w * (2 + \zeta_y - \zeta_z) - \zeta_z)/2$	$(\zeta_y + w * (2 + \zeta_y - \zeta_z) - \zeta_z)/2$	$(-\zeta_y + w * (2 + \zeta_y - \zeta_z) + \zeta_z)/2$
7C	$\chi^{S,R}$	$\zeta_z/4$	$1 + \zeta_x + ((-1+w) * \zeta_z)/2$	$1 + \zeta_y + ((-1+w) * \zeta_z)/2$	$(-2 + \zeta_z + w * \zeta_z)/2$

Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8A	$\chi^{S,L}$	$-\zeta_z/4$	$1 + \zeta_x - ((1+w) * \zeta_z)/2$	$1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
8A	$\chi^{S,LM}$	$(-\zeta_y + \zeta_z)/4$	$(2 + 2 * \zeta_x - (1+w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
8A	$\chi^{S,RM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_y)/2$	$(-2 - (1+w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$-1 - ((1+w) * \zeta_x)/2 + ((-1+w) * \zeta_y)/2 + \zeta_z$
8A	$\chi^{S,R}$	$(2 + \zeta_x)/4$	$(\zeta_x + w * (2 + \zeta_x))/2$	$w - \zeta_x/2 + (w * \zeta_x)/2 + \zeta_y$	$w - \zeta_x/2 + (w * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8B	$\chi^{S,L}$	$-\zeta_y/4$	$1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$1 - ((1+w) * \zeta_y)/2 + \zeta_z$
8B	$\chi^{S,LM}$	$(\zeta_y - \zeta_z)/4$	$(2 + 2 * \zeta_x + (-1+w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1+w) * \zeta_y - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
8B	$\chi^{S,RM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
8B	$\chi^{S,R}$	$(2 + \zeta_x)/4$	$(\zeta_x + w * (2 + \zeta_x))/2$	$w - \zeta_x/2 + (w * \zeta_x)/2 + \zeta_y$	$w - \zeta_x/2 + (w * \zeta_x)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8C	$\chi^{S,L}$	$-\zeta_z/4$	$1 + \zeta_x - ((1+w) * \zeta_z)/2$	$1 + \zeta_y - ((1+w) * \zeta_z)/2$	$(2 + \zeta_z - w * \zeta_z)/2$
8C	$\chi^{S,LM}$	$(-\zeta_x + \zeta_z)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_z)/2$	$(2 - (1+w) * \zeta_x + 2 * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(-2 - (1+w) * \zeta_x + \zeta_z + w * \zeta_z)/2$
8C	$\chi^{S,RM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_y)/2$	$(2 + (-1+w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$-1 + ((-1+w) * \zeta_x)/2 - ((1+w) * \zeta_y)/2 + \zeta_z$
8C	$\chi^{S,R}$	$(2 + \zeta_y)/4$	$w + \zeta_x - \zeta_y/2 + (w * \zeta_y)/2$	$(\zeta_y + w * (2 + \zeta_y))/2$	$w - \zeta_y/2 + (w * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8D	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$1 - ((1+w) * \zeta_x)/2 + \zeta_z$
8D	$\chi^{S,LM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
8D	$\chi^{S,RM}$	$(-\zeta_y + \zeta_z)/4$	$(-2 + 2 * \zeta_x - (1+w) * \zeta_y - \zeta_z + w * \zeta_z)/2$	$(2 + \zeta_y - w * \zeta_y + (-1+w) * \zeta_z)/2$	$(-2 - (1+w) * \zeta_y + \zeta_z + w * \zeta_z)/2$
8D	$\chi^{S,R}$	$(2 + \zeta_y)/4$	$w + \zeta_x - \zeta_y/2 + (w * \zeta_y)/2$	$(\zeta_y + w * (2 + \zeta_y))/2$	$w - \zeta_y/2 + (w * \zeta_y)/2 + \zeta_z$
Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8E	$\chi^{S,L}$	$-\zeta_x/4$	$(2 + \zeta_x - w * \zeta_x)/2$	$1 - ((1+w) * \zeta_x)/2 + \zeta_y$	$1 - ((1+w) * \zeta_x)/2 + \zeta_z$
8E	$\chi^{S,LM}$	$(\zeta_x - \zeta_y)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_y)/2$	$(2 + (-1+w) * \zeta_x + \zeta_y - w * \zeta_y)/2$	$1 + ((-1+w) * \zeta_x)/2 - ((1+w) * \zeta_y)/2 + \zeta_z$
8E	$\chi^{S,RM}$	$(\zeta_y - \zeta_z)/4$	$(-2 + 2 * \zeta_x + (-1+w) * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(-2 + (1+w) * \zeta_y - (1+w) * \zeta_z)/2$	$(2 + (-1+w) * \zeta_y + \zeta_z - w * \zeta_z)/2$
8E	$\chi^{S,R}$	$(2 + \zeta_z)/4$	$w + \zeta_x - \zeta_z/2 + (w * \zeta_z)/2$	$w + \zeta_y - \zeta_z/2 + (w * \zeta_z)/2$	$w + ((1+w) * \zeta_z)/2$
Region	Basis Function Parameters				
Basis Function	Coefficient	\arg_x	\arg_y	\arg_z	
8F	$\chi^{S,L}$	$-\zeta_y/4$	$1 + \zeta_x - ((1+w) * \zeta_y)/2$	$(2 + \zeta_y - w * \zeta_y)/2$	$1 - ((1+w) * \zeta_y)/2 + \zeta_z$
8F	$\chi^{S,LM}$	$(-\zeta_x + \zeta_y)/4$	$(2 + \zeta_x - w * \zeta_x + (-1+w) * \zeta_y)/2$	$(-2 - (1+w) * \zeta_x + \zeta_y + w * \zeta_y)/2$	$1 - ((1+w) * \zeta_x)/2 + ((-1+w) * \zeta_y)/2 + \zeta_z$
8F	$\chi^{S,RM}$	$(\zeta_x - \zeta_z)/4$	$(-2 + (1+w) * \zeta_x - (1+w) * \zeta_z)/2$	$(-2 + (-1+w) * \zeta_x + 2 * \zeta_y - \zeta_z - w * \zeta_z)/2$	$(2 + (-1+w) * \zeta_x + \zeta_z - w * \zeta_z)/2$
8F	$\chi^{S,R}$	$(2 + \zeta_z)/4$	$w + \zeta_x - \zeta_z/2 + (w * \zeta_z)/2$	$w + \zeta_y - \zeta_z/2 + (w * \zeta_z)/2$	$w + ((1+w) * \zeta_z)/2$

Table S7: Table describing how the shifting arguments (q_x, q_y, q_z) of the a coefficients vary by region.

Region S	Arguments of $a_\alpha(\mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z)$	$\mathbf{P} = \mathbf{L}$	$\mathbf{P} = \mathbf{LM}$	$\mathbf{P} = \mathbf{RM}$	$\mathbf{P} = \mathbf{R}$
	1A	(0, 0, 0)	(0, 0, 1)	(0, 1, 1)	(1, 1, 1)
	1B	(0, 0, 0)	(0, 1, 0)	(0, 1, 1)	(1, 1, 1)
	1C	(0, 0, 0)	(0, 0, 1)	(1, 0, 1)	(1, 1, 1)
	1D	(0, 0, 0)	(1, 0, 0)	(1, 0, 1)	(1, 1, 1)
	1E	(0, 0, 0)	(1, 0, 0)	(1, 1, 0)	(1, 1, 1)
	1F	(0, 0, 0)	(0, 1, 0)	(1, 1, 0)	(1, 1, 1)
	2E	(0, 0, -1)	(0, 0, 0)	(1, 0, 0)	(1, 1, 0)
	2F	(0, 0, -1)	(0, 0, 0)	(0, 1, 0)	(1, 1, 0)
	3A	(-1, 0, 0)	(0, 0, 0)	(0, 0, 1)	(0, 1, 1)
	3B	(-1, 0, 0)	(0, 0, 0)	(0, 1, 0)	(0, 1, 1)
	4B	(-1, 0, -1)	(-1, 0, 0)	(0, 0, 0)	(0, 1, 0)
	4F	(-1, 0, -1)	(0, 0, -1)	(0, 0, 0)	(0, 1, 0)
	5C	(0, -1, 0)	(0, 0, 0)	(0, 0, 1)	(1, 0, 1)
	5D	(0, -1, 0)	(0, 0, 0)	(1, 0, 0)	(1, 0, 1)
	6D	(0, -1, -1)	(0, -1, 0)	(0, 0, 0)	(1, 0, 0)
	6E	(0, -1, -1)	(0, 0, -1)	(0, 0, 0)	(1, 0, 0)
	7A	(-1, -1, 0)	(-1, 0, 0)	(0, 0, 0)	(0, 0, 1)
	7C	(-1, -1, 0)	(0, -1, 0)	(0, 0, 0)	(0, 0, 1)
	8A	(-1, -1, -1)	(-1, -1, 0)	(-1, 0, 0)	(0, 0, 0)
	8B	(-1, -1, -1)	(-1, 0, -1)	(-1, 0, 0)	(0, 0, 0)
	8C	(-1, -1, -1)	(-1, -1, 0)	(0, -1, 0)	(0, 0, 0)
	8D	(-1, -1, -1)	(0, -1, -1)	(0, -1, 0)	(0, 0, 0)
	8E	(-1, -1, -1)	(0, -1, -1)	(0, 0, -1)	(0, 0, 0)
	8F	(-1, -1, -1)	(-1, 0, -1)	(0, 0, -1)	(0, 0, 0)